Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1988

Design methodology using the Genesil Silicon Compiler.

Settle, Robert Howard.

http://hdl.handle.net/10945/23012

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

S41862

DESIGN METHODOLOGY USING THE
GENESIL SILICON COMPILER

by

Robert Howard Settle

September 1988

Thesis Advisor: Herschel H. Loomis, Jr.

Approved for public release; distribution is unlimited

T242345

REPORT DOCUMENTATION PAGE

a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED
1b RESTRICTIVE MARKINGS:
a. SECURITY CLASSIFICATION AUTHORITY:
3 DISTRIBUTION/AVAILABILITY OF REPORT:
b. DECLASSIFICATION/DOWNGRADING SCHEDULE:
. PERFORMING ORGANIZATION REPORT NUMBER(S):
5 MONITORING ORGANIZATION REPORT NUMBER(S):
a. NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School
6b OFFICE SYMBOL (*If applicable*): 62
7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate SChool
c. ADDRESS (*City, State, and ZIP Code*): Monterey, California 93943-5000
7b. ADDRESS (*City, State, and ZIP Code*): Monterey, California 93943-5000
a. NAME OF FUNDING/SPONSORING ORGANIZATION:
8b OFFICE SYMBOL (*If applicable*):
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:
c. ADDRESS (*City, State, and ZIP Code*):

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO.

1. TITLE (*Include Security Classification*): DESIGN METHODOLOGY USING THE GENESIL SILICON COMPILER
2. PERSONAL AUTHOR(S): SETTLE, Robert H.
3a. TYPE OF REPORT: Master's Thesis
13b TIME COVERED: FROM ________ TO ________
14. DATE OF REPORT (*Year, Month, Day*): September 1988
15 PAGE COUNT: 182
6. SUPPLEMENTARY NOTATION: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

7. COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*): Genesil Silicon Compiler design methodology; Custom 16 bit pipelined adder and multiplier performance; Top-down chip-design

9. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

The applications of silicon compilers, and the design methodology of the Genesil Silicon Compiler are described. The performance of Genesil system library adders and multipliers are compared with comparable custom pipelined adder and multiplier circuits built on the Genesil Silicon Compiler. High performance pipeline methods are discussed. The appendix is a tutorial illsutrating a Genesil system hierarchical top-down chip design, including simulation and timing analysis procedures.

0 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS
21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
2a NAME OF RESPONSIBLE INDIVIDUAL: LOOMIS, Herschel H. Jr.
22b TELEPHONE (*Include Area Code*): 408-646-3124
22c OFFICE SYMBOL: 62Lm

D FORM 1473, 84 MAR
83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office 1986—606-243

Approved for public release; distribution is unlimited

Design Methodology Using the Genesil Silicon Compiler

by

Robert Howard Settle
Lieutenant Colonel, United States Marine Corps
B.S., United States Naval Academy, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1988

ABSTRACT

The applications of silicon compilers, and the design methodology of the Genesil Silicon Compiler are described. The performance of Genesil system library adders and multipliers are compared with comparable custom pipelined adder and multiplier circuits built on the Genesil Silicon Compiler. High performance pipeline methods are discussed. The appendix is a tutorial illustrating a Genesil system hierarchical top-down chip design, including simulation and timing analysis procedures.

C.1

TABLE OF CONTENTS

I. INTRODUCTION.....1

II. GENESIL SILICON COMPILER.....5

- A. INTRODUCTION.....5
- B. ASIC DESIGN.....5
- C. GENESIL SYSTEM DESCRIPTION.....6

III. ADDER CIRCUITS.....23

- A. PIPELINED CIRCUITS FOR HIGH PERFORMANCE.....23
- B. FULL ADDER DESIGN.....24
- C. FOUR BIT ADDER DESIGN.....32
- D. SIXTEEN BIT ADDER DESIGN.....41
- E. PERFORMANCE SUMMARY.....61

IV. MULTIPLIER CIRCUITS.....62

- A. INTRODUCTION.....62
- B. 4, 8, AND 16 BIT GENESIL LIBRARY MULTIPLIER..64
- C. 4 BIT PIPELINED MULTIPLIER.....68
- D. 16 BIT PIPELINED MULTIPLIER.....75

V. CONCLUSIONS.....82

- A. SUMMARY.....82
- B. RECOMMENDATIONS.....83

APPENDIX.....84

LIST OF REFERENCES.....174

INITIAL DISTRIBUTION LIST.....176

I. INTRODUCTION

As integrated circuits (I.C.) grow increasingly complex, new methods are needed to manage the complexity, cost, and time consumed when designing and testing the desired I.C. system. There is a demand for a quick, and relatively economical design process to precede the actual chip layout. To meet this criteria, the methodology must have the capability of design at higher levels to specify, design, and simulate the desired circuit. A state-of-the-art solution to this requirement is the silicon compiler.

Loosely defined, a silicon compiler is a system which generates I.C. layouts from high-level descriptions. Originally, silicon compilation referred to a design methodology in lieu of a system or set of processes. The silicon compiler was analogous to compilation of machine code from a high-level language. An object was a graphic image rather than a block of executable code. Geometrics of the desired chip were constructed the same way as machine code, i.e., compiled from high-level languages. Early silicon compilers use "C" and "LISP" compilers. [Ref. 1]

The latest silicon compilers are complete design systems. Compilation is used as one mechanism for overall chip design. State-of-the-art silicon compilers have computer aided engineering (CAE) / computer aided design (CAD) as in the past, but also include geometry editing, symbolic editing, simulation, automatic placement, automatic routing, compaction, and design rule checking [Ref. 1].

Each state of a complex custom I.C. system design, from concept to silicon testing, requires a team of experts in each field. Each team normally is not an expert in the other areas of chip development. Team expertise is a necessary condition in the fields of requirements generation, logic implementation, circuit simulation, chip layout, and testing. Chip level silicon compilation now allows a systems engineering to design Very Large Scale Integrated (VLSI) chips. With a silicon compiler, the design is accomplished by using a top-down, hierarchical design methodology starting with a partitioned chip set, proceeding down to individual chips, modules, and finally blocks. The blocks, or bottom-level design elements, include various types of logic blocks including ALU's, PLA's, RAM's, ROM's, multipliers, and inverters [Ref. 2].

Generally, far less time is required to design a circuit with a silicon compiler than is necessary for a comparable manual/CAD design method using graphic layout tools. The silicon compiler makes possible the rapid, real time development and testing of a system. This is advantageous for designing and producing relatively small numbers of chips. This is especially attractive for military applications where small numbers of chips are required (hundreds and thousands vs. millions) and a rapid turnaround time is desired [Ref. 3].

Reference [1] contains a directory with capability comparisons of the silicon compilers currently available from commercial sources. The most notable systems observed in the directory concerning flexibility and overall performance were the Concorde Silicon Compiler, made by the Seattle Silicon Corp. and the Genesil Silicon Compiler produced by Silicon Compiler Corp.

Currently, the Naval Postgraduate School has the capability of VLSI design using full custom methods, the MacPitts Silicon Compiler, and the Genesil Silicon Compiler. Both the full custom and MacPitts methods depend on separate, time consuming, programming for simulation and timing analysis of a VLSI chip. In Genesil, simulation and timing analysis are integrated into the system.

Chapter II describes the Genesil System's stand alone capabilities for the design of a VLSI system. Chapter III briefly describes system pipelining theory, a comparison of Genesil library versus custom adders, followed by the design and performance results of a pipelined 16 bit adder built on the Genesil Silicon Compiler. Chapter IV contains performance comparisons of Genesil library multipliers versus custom multipliers, followed by the design and performance results of a custom 4 bit pipelined Wallace Tree structured multiplier, concluding with the design and performance results of a custom pipelined 16 bit parallel multiplier. The Appendix contains a tutorial for a top-down VLSI chip design for the Genesil Silicon Compiler.

II. GENESIL SILICON COMPILER

A. INTRODUCTION

The Genesil Silicon Compiler is based on silicon compilation, which is an Application Specific Integrated Circuit (ASIC) design method. ASIC design methodology also includes full custom, gate array, and standard cell methods. Full custom design is accomplished by a team of IC experts, whereas gate array, standard cell, and silicon compilation are based on the premise that the designer is not an IC expert [Ref. 4].

B. ASIC DESIGN

Full custom design is normally used by IC manufacturers producing vast quantities (millions) of standard off-the-shelf type chips, such as microprocessors. The chip is normally very dense with a full set of masks and customized probe cards for production tests since the designer and user are most likely not the same [Ref. 2]. Full custom design is time consuming and expensive, which can be attributed to the complexity of the design parameters for high density chips. Design parameters, at the full custom level, are a constant tradeoff involving performance (speed, power, function), die size, ease of test generation, and testability [Ref. 5].

Gate array design is accomplished by interconnecting the appropriate rows and columns of transistors with metal layers defining the circuit from appropriate netlist libraries. The array is prefabricated and a circuit design is "fitted" to the array. Processing time is low, but circuit density is also low. Gate array vendors provide macros containing predefined patterns to form SSI circuits such as NAND and NOR gates [Ref.4]. This presents problems when attempting to translate a high level specification from one vendor to another.

Standard cell design is based upon the same methodology as the gate array design except it differs in the manufacture cycle. The gate array is a pre-manufactured wafer to which metal is added to form the IC. The standard cell does not have pre-defined transistor locations. The manufacturing process is similar to full custom, requiring all layers to be created. This does result, however, in a more dense circuit than that produced by the gate array method.

Silicon compilation produces a circuit which is very similar to a full custom design by synthesizing the circuit with a top-down, hierarchical design methodology consisting of chip sets, individual chips, modules and blocks [Ref. 5]. Silicon compilation provides the interface between high level design specifications and a variety of design tools which produce efficient IC layouts.

C. GENESIL SYSTEM DESCRIPTION

The Genesil Silicon Compiler System is a design automation system which provides the user with the capability of designing VLSI circuits from high level system description to manufacture tapeout by producing the IC circuits from architectural descriptions. The system is composed of menus, commands, and forms used in the following activities descriptions. The system uses the UNIX Operating System. A Genesil System Overview is shown in Figure 2.1 [Ref. 6:p. 2.3].

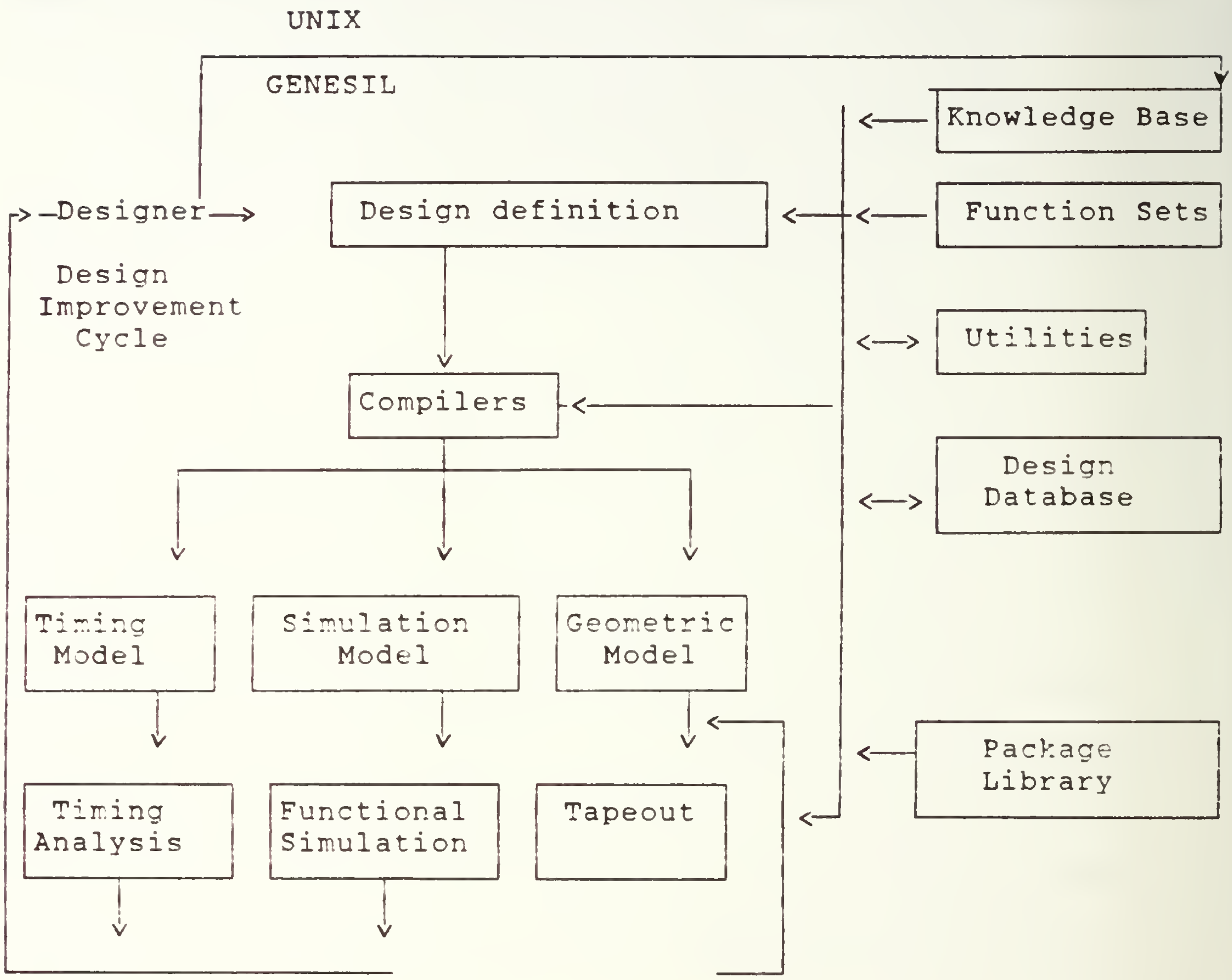


Figure 2.1 Genesil Overview

1. Methodology

Prior to beginning a Genesil session, the user should pre-plan all functional and performance specifications. With this initial plan, the user is able to rapidly observe exploratory ("first cut") designs of the required specifications. After as many alterations as desired or required, the detailed design can be completed to include simulation and timing analysis. Next, the physical design process is "invoked" by using the Floor planning feature. Once floor planned, verification is again conducted by functional simulation for logic, and timing analysis for performance. The chip is then ready for manufacture interface. The Genesil methodology is illustrated in Figure 2.2 [Ref. 6:p. 2.8].

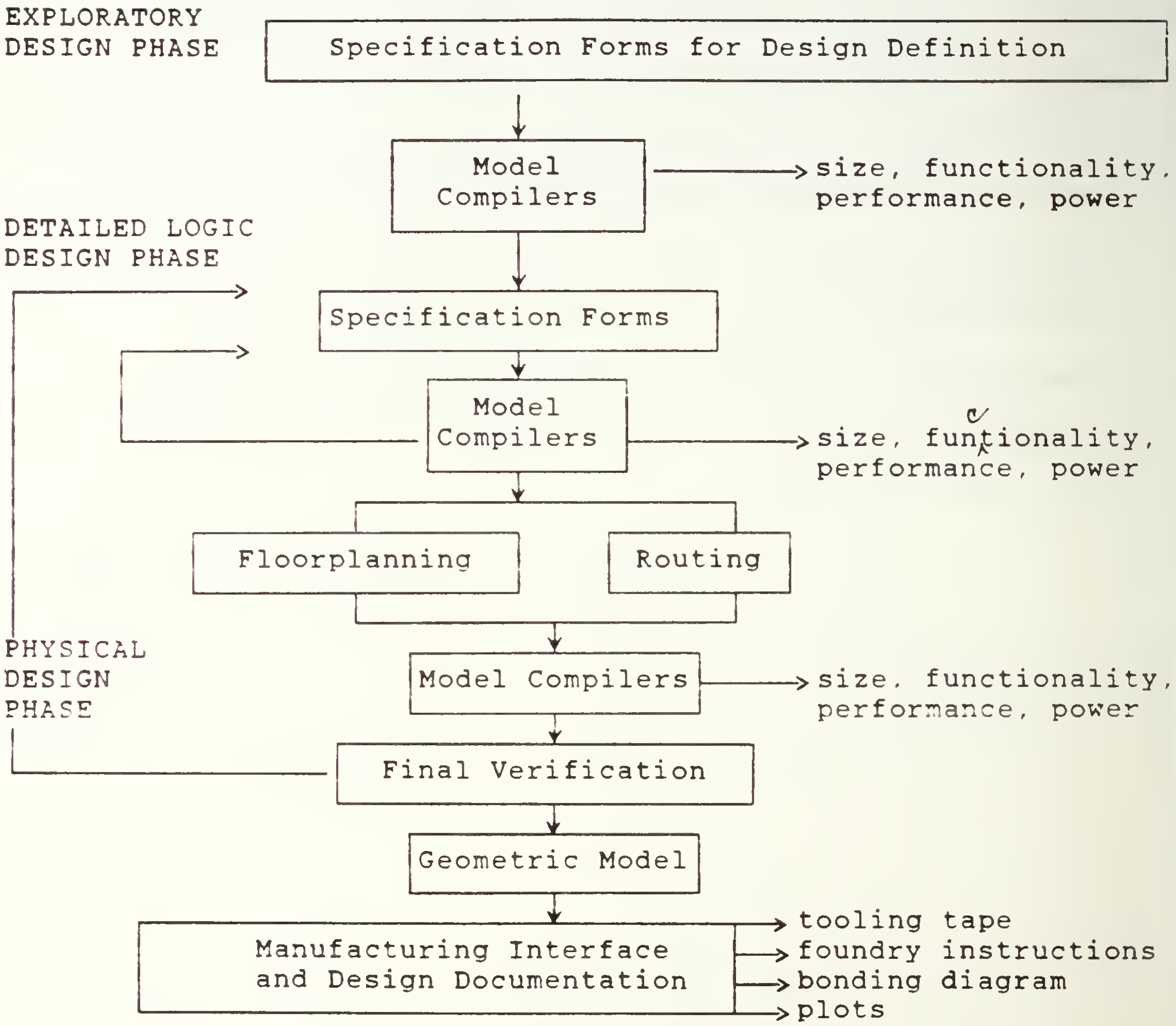

EXPLORATORY DESIGN PHASE
Specification Forms for Design Definition
Model Compilers
size, functionality, performance, power
DETAILED LOGIC DESIGN PHASE
Specification Forms
Model Compilers
c
size, funtionality, performance, power
Floorplanning
Routing
PHYSICAL DESIGN PHASE
Model Compilers
size, functionality, performance, power
Final Verification
Geometric Model
Manufacturing Interface and Design Documentation
tooling tape
foundry instructions
bonding diagram
plots

Figure 2.2 Genesil Methodology

2. Design

Genesil is an object oriented system which uses a hierarchical pathname system based on the UNIX operating system pathnames. Objects are selected, attached, detached, moved, and removed from the user's account. Objects include blocks, modules, chips, and chip-sets.

Blocks are the lowest level objects in the Genesil System object hierarchy. Blocks are created by the system block generator as directed by the user's functional specifications. There are three types of blocks, independent blocks, data-path blocks, and random logic blocks. Independent blocks include complex stand alone blocks of logic such as ROM's and PLA's. Data-path blocks are designed specifically for functions that manipulate parallel data. Random logic blocks contain user specified gate level logic. Modules are a collection of blocks, other modules (submodules), and parallel data-path modules [Ref. 7:p. 1.2].

Modules are intermediate objects in the hierarchy. Modules are a collection of blocks and other modules (submodules) [Ref. 7:p. 1.2].

Chips are complete integrated circuits. Chips contain blocks, modules, pad specifications, interconnection lists, positioning information for blocks and modules and packaging information. Chipsets are a collection of chips. Chipsets include chip interconnection lists, and user-supplied simulation model programs and timing analysis models [Ref. 6:p. G.4].

After an object is attached from the SELECT_OBJECT menu, a Header Form is completed which specifies function type (i.e., RAMs, PLAs, random logic, etc.) and fabline. Next, a Specification Form is used to implement detailed information about the object. The Specification Form varies, depending on the object type selected. The Specification Form is the heart of the design process. This is where all design specifications are designated by the user. Functional objects are attached/detached, signals attached/detached, and bus widths designated. Once the Specification Form is completed to the user's specifications, the system generates a form check which identifies any incorrect signal connections which can be corrected immediately. Figure 2.3 [Ref. 6:p. 5.16] illustrates the design description hierarchy.

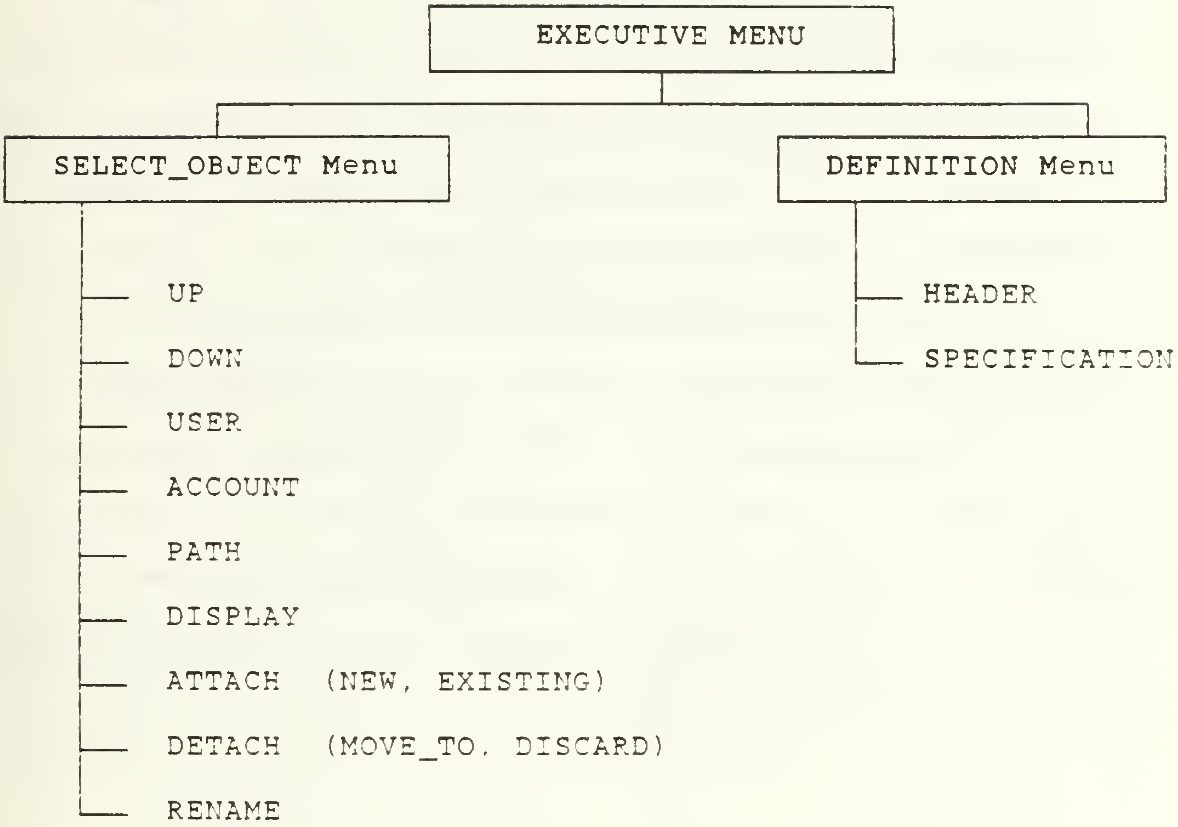


Figure 2.3 Design Description Hierarchy

3. Netlisting

A net is a connection between two or more objects. Objects are any one or combination of blocks, modules. chips, or chipsets. A netlist is a listing of all the nets in a module or chip. The netlisting feature allows the user to specify the interconnections between objects. Object-netlisting and Net-netlisting are two options which may be selected and show the same information from different points of reference. The Object-netlist form is used to define connections of sub-objects to the system net. The Net-netlist form is used to specify signal names to be combined into a network which the system creates. Used in this context, a signal is synononomous with a node, from which a single node is formed from all connectors and nets which are electrically equivalent. Netlisting can also be accomplished between chips to form chipsets. Figure 2.4 shows the netlisting commands hierarchy [Ref. 6:p. 6.18].

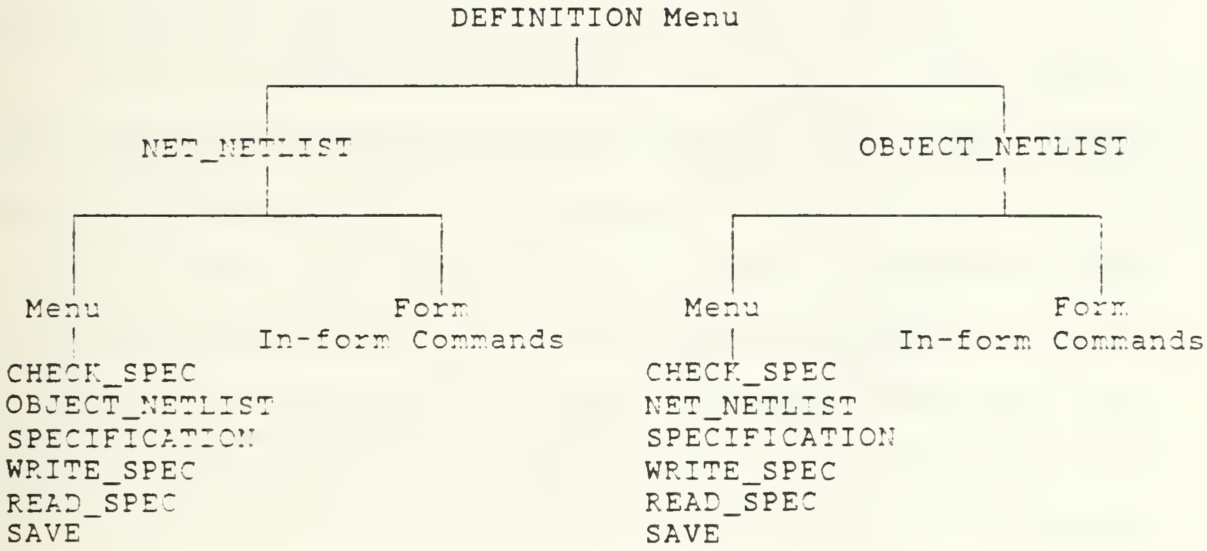


Figure 2.4 Netlisting Command Hierarchy

4. Floorplanning

Floorplanning consists of the actual placement of objects on a module or chip, the connection of pins to pads or pinout, and fusion order specification. Placement refers to the actual geographic locations of objects in relation to one another. The placement feature allows the user to graphically determine the placement between objects to minimize wire lengths. Genesil provides the appropriate menu depending on the object type. The pinout option provides for assignment of external signals to on chip (signals not local to an object) and off chip (I/O pins and pads) and is required for all modules and chips. Fusion allows the user to create and modify routing channel assignments on the floorplan by binding objects together to form channels [Ref. 6:p. 7.15]. Fusion may be accomplished automatically and then changed manually from the command selection form for more efficient routing. Figure 2.5 [Ref. 6:p. 7.25] depicts the Floorplanning Command Hierarchy.

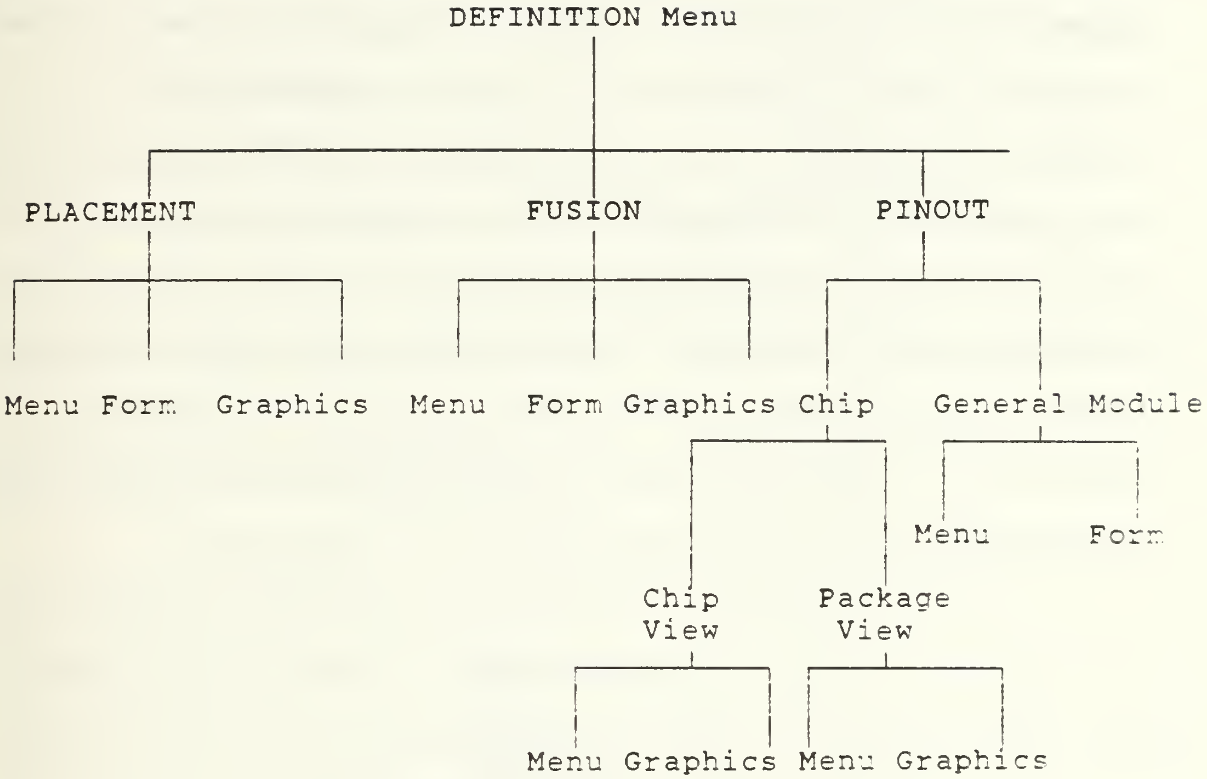


Figure 2.5 Floorplanning Command Hierarchy

5. Compiling

Compile is a command option that allows the user to force an immediate compilation of the complete set of views or selected views of the selected object. A view is one of three Genesil System representations of a block, which are geometric, functional, and timing. Compilation must be completed prior to simulation, timing analysis, plot, or tooling activities. It is automatically done if simulation or timing analysis is attempted prior to compile selection because the system checks the objects files for compilation currency. For efficiency, it was found that each block of objects should be compiled at the completion of design and netlisting. Figure 2.6 [Ref. 8:p. A.7] shows the compile menu of commands.

```
COMPILE Menu
       |
    SIM_MODEL
    TA_MODEL
    LOAD_MODEL
    GATE_MODEL
    LAYOUT
    BUILD_ALL
    CHECK
    AUTO_DEF_SPEC
           |
       INTERACTIVE
       ABORT_GO_ON
```

Figure 2.6 Compile Menu

During initial design phase, the most significant compile commands include Build_All, Sim_Model, and Ta_Model. The Build_All commands compile all views of the current object. Sim-Model compiles the simulation model needed for simulation of the current model. Ta-Model compiles the timing model necessary for timing analysis of the current object.

6. Simulation

The purpose of simulation is to verify that the design and layout generated by the current object are logically correct. The layout is synonomous with the geometric view of an object and is the physical equivalent of that object. The two most significant modes are functional and switch-level simulation.

Functional simulation generates a gate-level model for general purpose simulation and is independent of technology and layout. It is dependent only on circuit functionality and input signal changes. To perform functional simulation, the block must be defined and netlisted. Functional simulation is based on a demand-evaluation algorithm which creates a functional model that simulates only the minimum logic required for correct results [Ref. 9:p. 2.1]. The user designates net values and manually advances time across a clock edge.

Following design functionality verification, the object is floorplanned, resulting in the compilation of a layout. From the layout, switch-level simulation is used to verify functionality of the actual layout of the chip. Switch-level simulation uses an event-driven algorithm which is 5-10 times slower than functional simulation and is best used during final verification only. The algorithm is much slower because it depends on detailed signal propagation data extracted from timing analysis [Ref. 9:p. 2.2].

7. Timing Analysis

Timing command selection places the user in the general Timing Analyser (TA). The system then uses an algorithm that requires no test vectors and generates the following timing specs [Ref. 10:p. 1.1]:

Object propagation delays
Paths limiting clock frequency
Duty-cycle constraints
Input setup and hold times
Output delays
Internal node setup, hold times, and signal delays
Path delays between internal nodes.

The user then selects a menu command to generate a timing report for the desired information listed above. The TA can be used at any time during design following Definition Specification for random logic objects or blocks, and following Definition Specification and Floorplanning for modules or chips. Figure 2.7 [Ref. 10:p.1.1] shows the Timing Analysis environment.

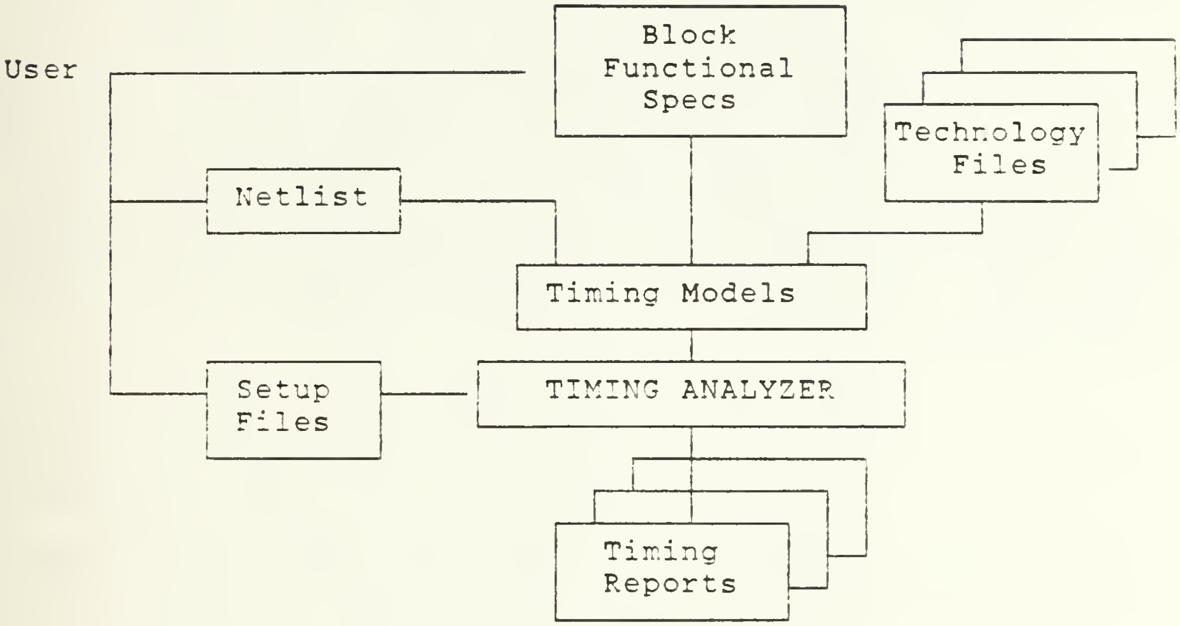


Figure 2.7 Timing Analysis Environment

8. Manufacture Interface

After a chip is completely specified and verified in terms of functionality, timing, power dissipation, and size, the design is ready to be sent to a foundry specified on the chip's definition header form. A foundry, or factory which produces the chip, may be changed at any time by changing the selection on the header form and re-compiling the chip. Design specifications are altered with each foundry change. The fabline, or process used to make the chip, changes with each foundry change which affects chip size, power, and timing results. Changes occur because chip layout differs due to the selected fabline's design rule check, and each foundry has its own models for devices built on that particular fabline [Ref. 6:p. 11.2].

III. ADDER CIRCUITS

A. PIPLINED CIRCUITS FOR HIGH PERFORMANCE

The purpose of pipelined circuits is to increase the through-put or performance of a circuit by splitting the task to be performed into cascaded sub-functions or smaller pieces and allocating separate hardware to each piece. Each piece or sub-function is defined as a stage. A stage normally consists of two components. They are the combinational logic to perform the sub-function, and a latch or flip-flop to save the output of one stage as input to the next. The concept is analagous with a physical pipeline or automobile assemblyline. Data flows through the stages of the circuit at a rate which is independent of the length of the pipeline or number of stages. If the overall function is completed in "X" nanoseconds(ns), and the function is divided into "N" stages, or sub-functions, then the output of the original function can theoretically be increased by "X/N" ns. This results in an "N" fold increase of performance [Ref. 11].

There are physical limitations on "N" due to hardware technology, the function being pipelined, clock-skew, and critical race. References 11 and 12 address hardware and function limitations. Clock-skew, due to circuit lengths, loading, and driver circuits, makes it nearly impossible to guarantee that all stages of a pipelined circuit receive the same pulse at exactly the same time. Critical race refers to the situation where a datapath through a logic block in a stage may be so short that, if the latch changes its output early, the change may reach the next staging latch and change it during the same clock pulse [Ref. 1]. For this reason, usually flip-flops are used between logic stages or a two phase clock system is used with latches, each phase serving alternate stage latches. Two stages and a basic pipeline clock are shown in Figure 3.1 [Ref. 12:p. 36].

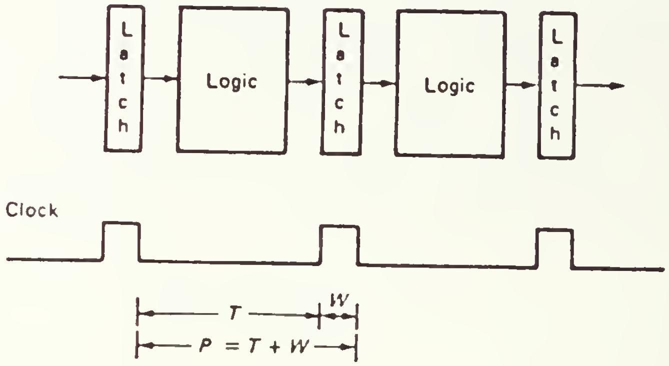


Figure 3.1 Two Stages and Pipeline Clock

B. FULL ADDER DESIGN

1. Introduction

Full adders serve a significant role in high performance pipelined signal processing circuits. High performance custom signal processing filters consist of pipelined adders and multipliers built from full adders. The Genesil Silicon Compiler system library contains full adders which can be programmed, via a menu, from 1 to 16 bits in width. The performance and size of a 1 bit library full adder were compared with the performance and size of a custom full adder built on the Genesil System. Motivation for performance and size data stemmed from the research data presented in Chapter IV, where full adders were used to build high performance pipelined multiplier chips.

2. Genesil Silicon Compiler Library Full Adder

A full adder, shown in Figure 3.2 [Ref, 13;p. 2.1], was extracted from the Genesil Compiler Library, Volume III, which is a collection of all system random logic blocks available. The figure illustrated the only transparent information available to the user concerning a full adder.

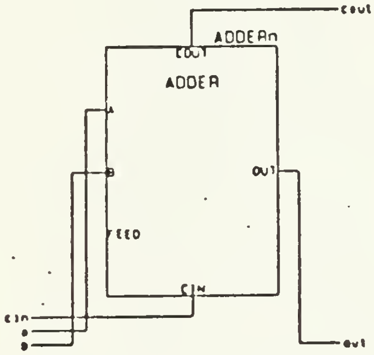


Figure 3.2 Genesil View of Adder Block

The figure depicts two data input buses (A and B), and a single carry input (Cin) which are added together resulting in the data output bus (OUT) and a carry output (COUT) [Ref. 3]. The manual offered no internal logic circuit diagrams nor performance and size specifications for the adder. Since the adder width can be varied from 1 to 16 bits, the interest was in whether the system used a general algorithm for adder construction, possibly resulting in wasted size and unnecessary hardware, or if it "customized" the adder to user specifications.

A 1 bit full adder block called lib_fulladd_blk was constructed, and a VLSI layout is shown in Figure 3.3.

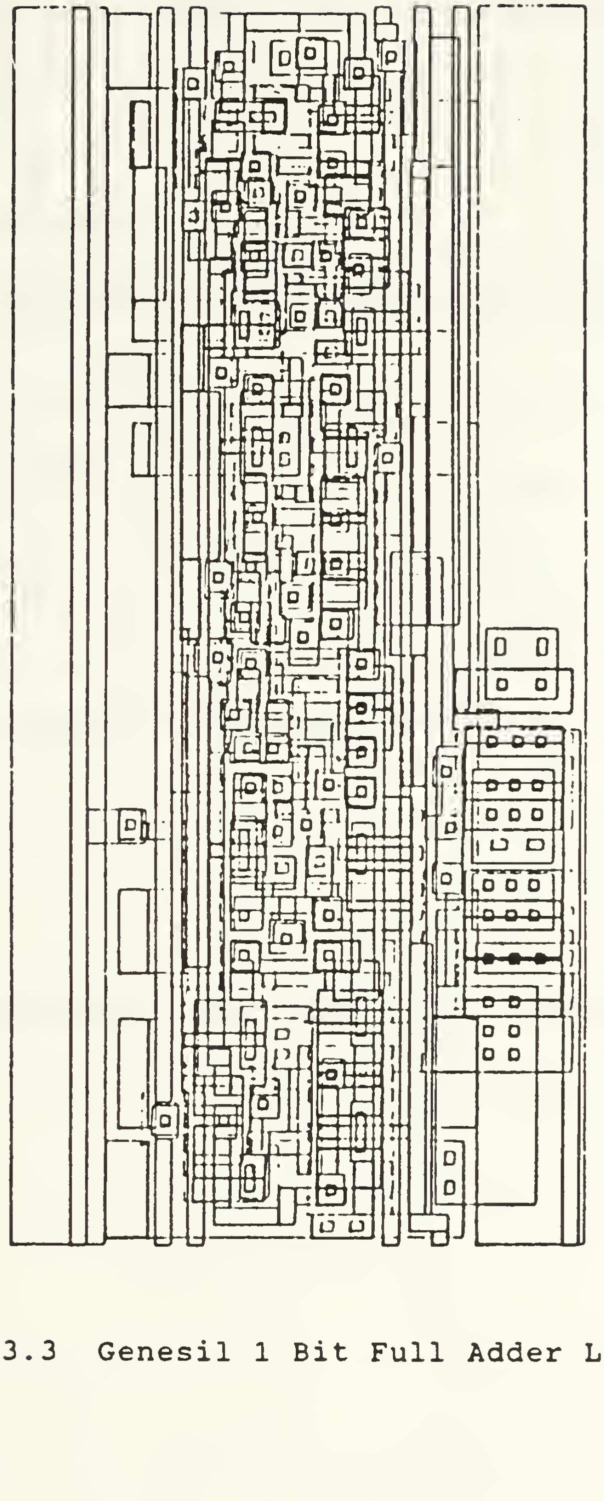


Figure 3.3 Genesil 1 Bit Full Adder Layout

The object size, calculated by the system, was 3.56 X 7.42 mils, resulting in a total area of 26.15 square mils.

Timing analysis was performed on the block which resulted in the Timing Analyser output propagation delays shown in Table I.

TABLE I
GENESIL FULL ADDER OUTPUT PROPAGATION DELAYS

OUTPUT	OUTPUT DELAYS (ns)	
	MIN	MAX
COUT	1.9	4.7
SUM	1.6	5.2

3. Custom Full Adder

A typical full adder was constructed from exclusive-or gates, AND gates, and OR gate as shown in Figure 3.4 [Ref. 14].

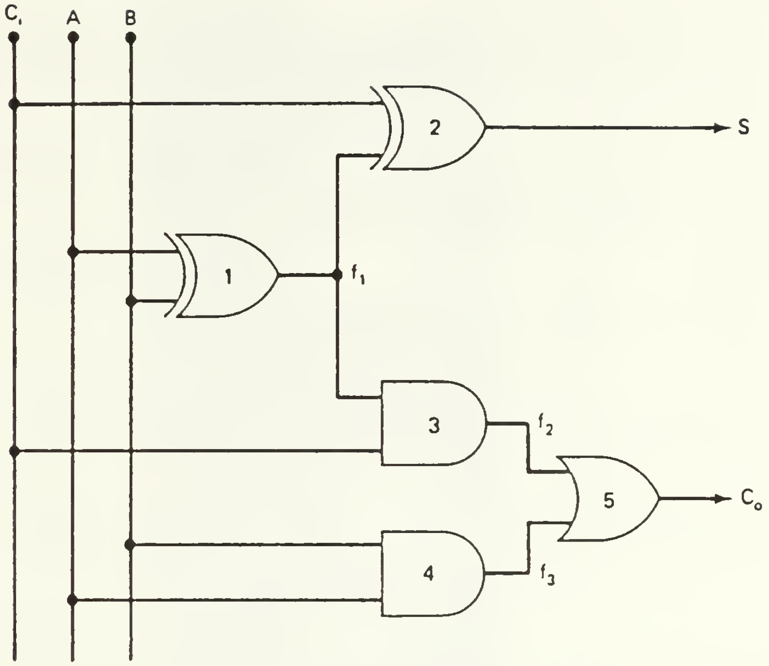


Figure 3.4 Full Adder Logic Circuit

An object called cus_fulladd_blk was constructed and a VLSI layout is shown in Figure 3.5.

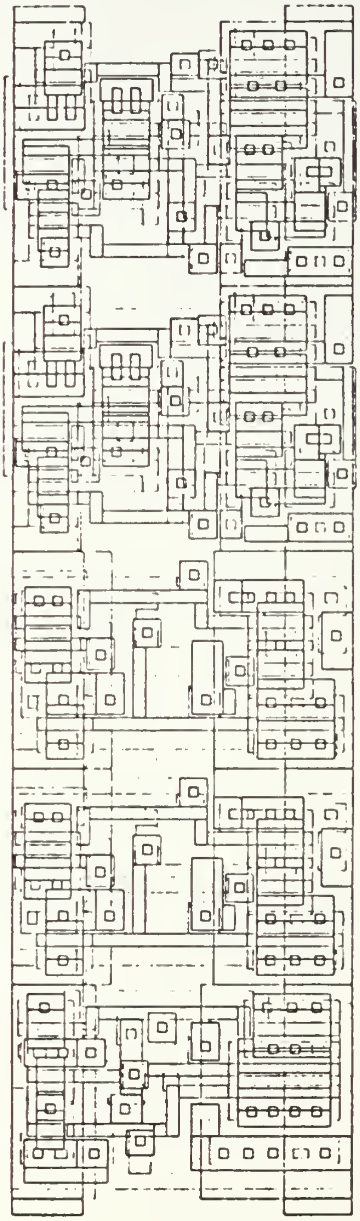


Figure 3.5 Custom 1 Bit Full Adder Layout

The object size, in mils, as calculated by the system was 7.58 X 2.42. This resulted in a total area of 18.34 square mils.

Timing analysis was performed by the system timing analyzer resulting in output propagation delays presented in Table II.

TABLE II
CUSTOM FULL ADDER OUTPUT PROPAGATION DELAYS

OUTPUT	OUTPUT DELAYS (ns)	
	MIN	MAX
COUT	2.5	5.0
SUM	1.4	4.6

4. Results

The results of a comparison between the system library full adder and the custom full adder are summarized in Table III.

TABLE III
SIZE AND PERFORMANCE SUMMARY

	MAX PROPAGATION DELAY (ns)	TOTAL AREA (SQ MILS)
GENESIL FULL ADDER	5.2	26.15
CUSTOM FULL ADDER	5.0	18.34

The data indicate that the custom full adder provides a 0.2ns performance improvement with 7.81 square mils saving in area.

C. FOUR BIT ADDER DESIGN

1. Introduction

The four bit adder is the building block for high performance pipelined adders. A basic four bit adder is a ripple carry circuit consisting of four full adders. The carry out of each adder ripples down as one of the three inputs to the next adder. Performance size considerations influence the appropriate design. Designs include pipelining full adders with latches, four bit carry-look-ahead adders (CLA), and pipelined CLA. A pure performance pipelined CLA is described in Reference 15.

2. Genesil Silicon Compiler Library Four Bit Adder

A library four bit adder called lib_4bit_blk was constructed by using the Random Logic Block Specification menu shown in Figure 3.6.

```
Block type:    ADDER
Block index:   0
Name:          >ADDER0__
Width:         >_4

                     Regime
Connector    Width    : Timing
A             4       1 Prop(t)      [ 3]       >A3__________
                                     [ 2]       >A2__________
                                     [ 1]       >A1__________
                                     [ 0]       >A0__________
B             4       1 Prop(t)      [ 3]       >B3__________
                                     [ 2]       >B2__________
                                     [ 1]       >B1__________
                                     [ 0]       >B0__________
OUT           4       1 Prop(t)      [ 3]       >S3__________
                                     [ 2]       >S2__________
                                     [ 1]       >S1__________
                                     [ 0]       >S0__________
CIN           1       1 Prop(t)      [ 0]       >Ci__________
COUT          1       1 Prop(t)      [ 0]       >Co__________
FEED          4       1 Feed thru    [ 3]       >FALSE_______
                                     [ 2]       >FALSE_______
                                     [ 1]       >FALSE_______
                                     [ 0]       >FALSE_______
```

Figure 3.6 Genesil 4 Bit Adder Specification Menu

Signal names were specified for the A and B buses, the 4 bit sum (OUT), carry-in (CIN), and carry-out (COUT).

A VLSI layout of the adder was constructed by using the system's plot feature and is shown in Figure 3.7.

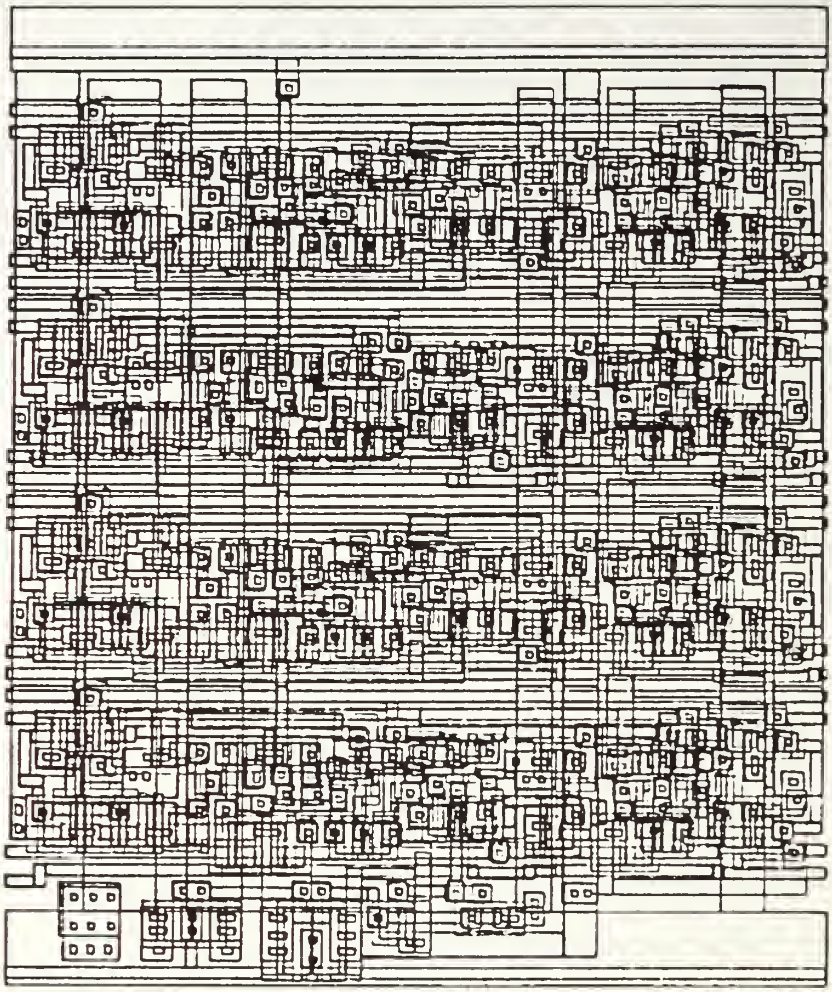


Figure 3.7 Genesil Layout lib_4bit_blk

The size of the adder, calculated by the system during layout, measured 8.77 X 7.42 mils, resulting in a total area of 65.07 square mils.

Timing analysis, provided by the System Timing Analyzer, produced the output propagation delays for all output signals. Data are provided in Table IV.

TABLE IV
GENESIL 4 BIT ADDER OUTPUT PROPAGATION DELAYS

OUTPUT	OUTPUT DELAYS (ns)	
	MIN	MAX
Co	1.9	9.5
S0	1.6	5.2
S1	3.1	6.3
S2	2.3	8.0
S3	3.1	9.2

Maximum propagation delay was 9.5 ns, occurring at the carry-out (Co) output.

3. Custom Four Bit Carry-Look-Ahead (CLA) Adder

The Genesil System manuals provide no information concerning any CLA features of the library adder. Without CLA, addition can become inefficient, but by pipelining conventional adder circuitry, performance can be increased at the price of additional hardware. CLA circuits involve more hardware than ripple carry circuits, but are faster.

The principle of CLA circuits involves anticipating when and where a carry will be generated. The circuit "looks ahead" to see where the carry is needed. References 15 and 16 provide detailed CLA algorithm development and analysis. The circuit shown in Figure 3.8 [Ref. 17] was constructed for a performance and size comparison with the Genesil 4 bit adder.

Figure 3.9 shows the Random Logic Functional Specification menu used to build the CLA circuit. Each random logic object was specified, signals designated, and the circuit net-listed.

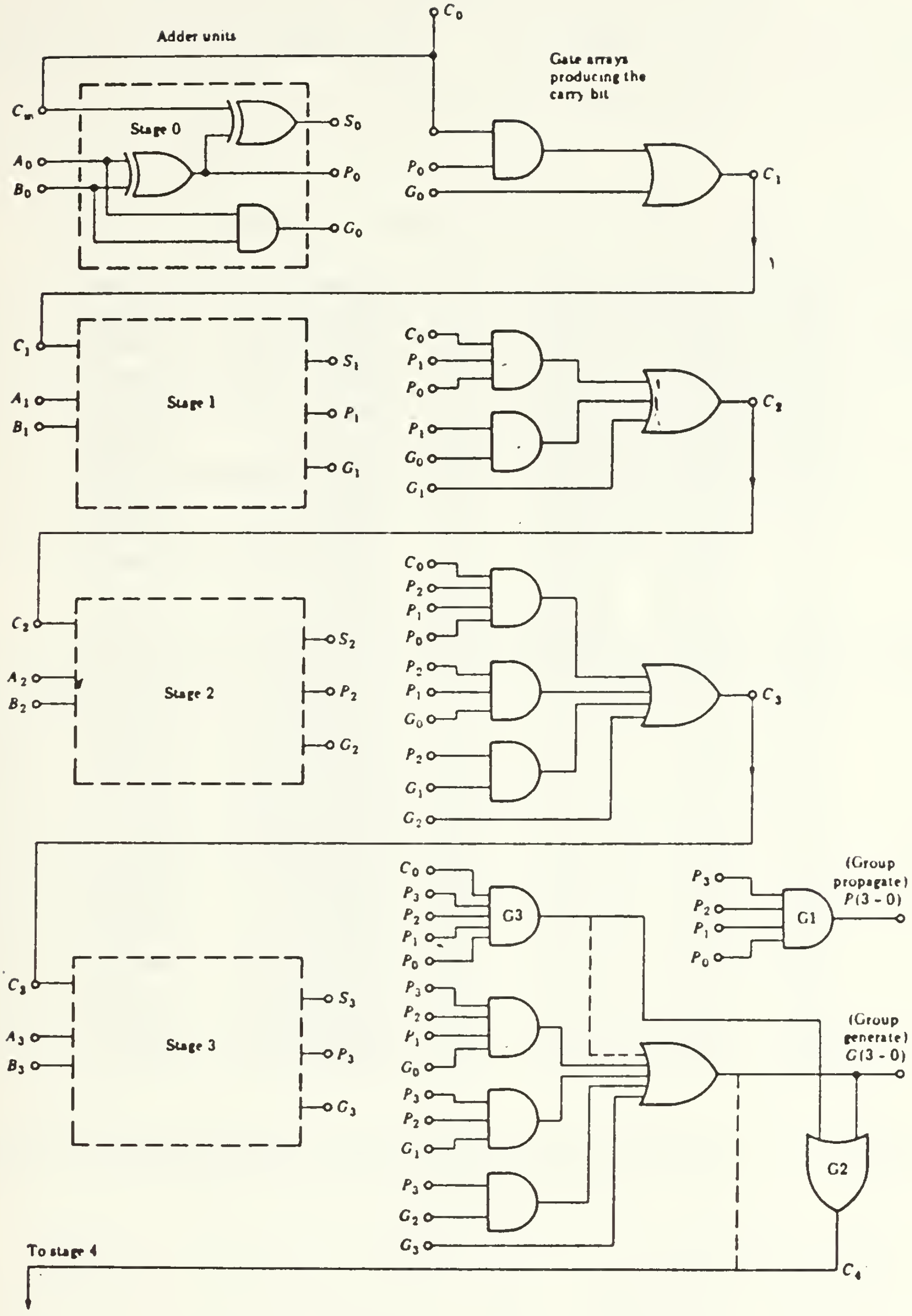


Figure 3.8 4 Bit CLA Adder

```
DEL   EDIT   MOVE    0:   >XOR0_____    (XOR)
DEL   EDIT   MOVE    1:   >XOR1_____    (XOR)
DEL   EDIT   MOVE    2:   >AND2_____    (AND)
DEL   EDIT   MOVE    3:   >AND3_____    (AND)
DEL   EDIT   MOVE    4:   >OR4______    (OR)
DEL   EDIT   MOVE    5:   >XOR5_____    (XOR)
DEL   EDIT   MOVE    6:   >XOR6_____    (XOR)
DEL   EDIT   MOVE    7:   >AND7_____    (AND)
DEL   EDIT   MOVE    8:   >AND8_____    (AND)
DEL   EDIT   MOVE    9:   >AND9_____    (AND)
DEL   EDIT   MOVE   10:   >AND10____    (AND)
DEL   EDIT   MOVE   11:   >OR11_____    (OR)
DEL   EDIT   MOVE   12:   >XOR12____    (XOR)
DEL   EDIT   MOVE   13:   >AND13____    (AND)
DEL   EDIT   MOVE   14:   >AND14____    (AND)
DEL   EDIT   MOVE   15:   >AND15____    (AND)
DEL   EDIT   MOVE   16:   >AND16____    (AND)
DEL   EDIT   MOVE   17:   >OR17_____    (OR)
DEL   EDIT   MOVE   18:   >XOR18____    (XOR)
DEL   EDIT   MOVE   19:   >XOR19____    (XOR)
DEL   EDIT   MOVE   20:   >AND20____    (AND)
DEL   EDIT   MOVE   21:   >AND21____    (AND)
DEL   EDIT   MOVE   22:   >AND22____    (AND)
DEL   EDIT   MOVE   23:   >AND23____    (AND)
DEL   EDIT   MOVE   24:   >AND24____    (AND)
DEL   EDIT   MOVE   25:   >OR25_____    (OR)
```

Figure 3.9 Random Logic Functional Specification Menu

An object layout of the CLA block called cus_cla4bit_blk was completed. The system calculated the size of the object as 44.53 X 2.42 mils, resulting in a total area of 107.76 square mils.

Since this was a custom object, the System Functional Simulator was used to test for correct logic. Random values were put on the inputs, the system clocks cycled, and results read from the Functional Simulator menu. An example simulation run extracted from the Functional Simulator is included as Figure 3.10.

```
> /cus_cla4bit_blk is of type genblock/rl with 28 ports
> port 0 I A0  to  NC = 1
> port 1 I A1  to  NC = 1
> port 2 I B0  to  NC = 1
> port 3 I A2  to  NC = 1
> port 4 I B1  to  NC = 1
> port 5 I A3  to  NC = 1
> port 6 I B2  to  NC = 1
> port 7 I B3  to  NC = 1
> port 8 O C4  to  NC = 1
> port 9 O S0  to  NC = 1
> port 10 O S1  to  NC = 1
> port 11 O S2  to  NC = 1
> port 12 O S3  to  NC = 1
> port 13 I Ci  to  NC = 1
```

Figure 3.10 4 Bit CLA Simulation

The System Timing Analyser was used for output propagation delays data for all output signals. The results are shown in Table V.

TABLE V
4 BIT CLA OUTPUT PROPAGATION DELAYS

OUTPUT	OUTPUT DELAYS (ns)	
	MIN	MAX
C4	2.8	6.9
S0	1.4	5.1
S1	3.8	7.6
S2	3.8	8.7
S3	3.5	9.0

Maximum propagation delay was 9.0 ns, occurring at the most significant bit output (S3).

4. Results

The results of the comparison are summarized in Table VI.

TABLE VI
SIZE AND PERFORMANCE SUMMARY

	MAX PROPAGATION DELAY (ns)	TOTAL AREA (Sq Mils)
GENESIL 4 BIT ADDER	9.5	65.07
CUSTOM 4 BIT CLA ADDER	9.0	107.76

The data indicate that the custom CLA provides a 0.5 ns performance improvement, but is 42.69 square mils larger than the library adder. The performance of the library adder indicates that the circuit probably has CLA circuitry included in the system adder algorithm.

D. SIXTEEN BIT ADDER DESIGN

1. Introduction

The pipelined 16 bit adder can be used in conjunction with two's complement hardware for special purpose signal processors, in the final stages of a 16 bit multiplier as presented in Chapter IV, or in various other capacities involved with high performance special purpose hardware. This section compares the performance and size of a 16 bit Genesil library adder with a custom 16 bit adder built on the Genesil System. Additional pipelined adder designs and performance data are available in Reference 18.

2. Genesil Library 16 Bit Adder

A library 16 bit adder called lib_16bit_blk was constructed by using the Random Logic Block Specification menu shown in Figure 3.11.

```
Block type:   ADDER
Block index:  O
Name:         >ADDERO__
Width:        >16

                 Regime
Connector Width     Timing
A          16     1 Prop(t)     >A[15:0]_____
B          16     1 Prop(t)     >B[15:0]_____
OUT        16     1 Prop(t)     >S[15:0]_____
CIN        1      1 Prop(t)     >Ci__________
COUT       1      1 Prop(t)     >Co__________
FEED       16     1 Feedthru    >FALSE*16
```

Figure 3.11 Genesil 16 Bit Adder Specification Menu

Input signal names, in bus notation, were specified for the A (A[15:0]) and B (B[15:0]) buses, and Cin. Output signals included sum (s[15:0]), in bus notation, and Co.

A VLSI layout from the system plot feature is included as Figure 3.12.

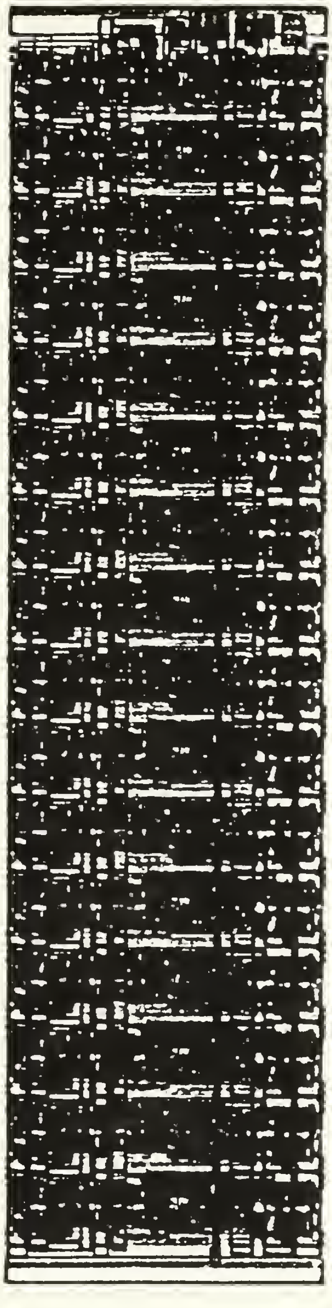


Figure 3.12 Genesil Layout lib_16bit_blk

The size of the library 16 bit full adder was calculated by the system to be 29.56 X 7.42 mils, resulting in a total area of 219.33 square mils.

Output propagation delays for all signals were calculated by the System Timing Analyser and are presented in Table VII.

TABLE VII
GENESIL 16 BIT ADDER OUTPUT PROPAGATION DELAYS

OUTPUT	OUTPUT DELAYS (ns)	
	MIN	MAX
Co	1.9	27.2
S[0]	1.6	5.2
S[10]	2.3	19.8
S[11]	3.1	21.1
S[12]	2.3	22.8
S[13]	3.1	24.0
S[14]	2.3	25.7
S[15]	3.1	27.0
S[1]	3.1	6.3
S[2]	2.3	8.0
S[3]	3.1	9.2
S[4]	2.3	11.0
S[5]	3.1	12.1
S[6]	2.3	13.9
S[7]	3.1	15.2
S[8]	2.3	16.9
S[9]	3.1	18.1

Maximum propagation delay was 27.2 ns, occurring at the carry-out (Co) signal.

3. Custom 16 Bit Pipelined Adder

A custom 16 bit adder was constructed using Genesil library 4 bit adders for the add logic, and 2-phase library D flip/flops to retain the data between each stage. The library adders were used because the performance and size comparison with a 4 bit CLA previously completed indicated that the differences were insignificant for the purposes of this section. An attempt to build a custom Earle latch, as presented in Reference 12, failed because the system disallowed random logic gate output signals to simultaneously perform as both external output signals and internal feedback signals. This feature is required for memory in both the D flip/flop and Earle latch. This problem was not pursued further because a library D flip/flop setup and hold time was found not to exceed 6.5 ns. This was approximately 3.0 ns less than the 4 bit library adder which was used in each stage of the adder. The adder logic, therefore, was the dominating delay factor driving the clock speed. Figure 3.13 shows the design, in block diagram form, used to construct the adder on Genesil.

Figure 3.14 shows a VLSI layout of the custom 16 bit pipelined adder module, without input/output pads, constructed for performance and size comparison with the library adder. The module was floorplanned using the list-best command in the floorplan menu. This feature graphically advised the user of the best placement of the 5 stages in the module for optimum routing and fusion. The adder module was 21.89 X 339.19 mils, resulting in a total area of 7424.87 square mils.

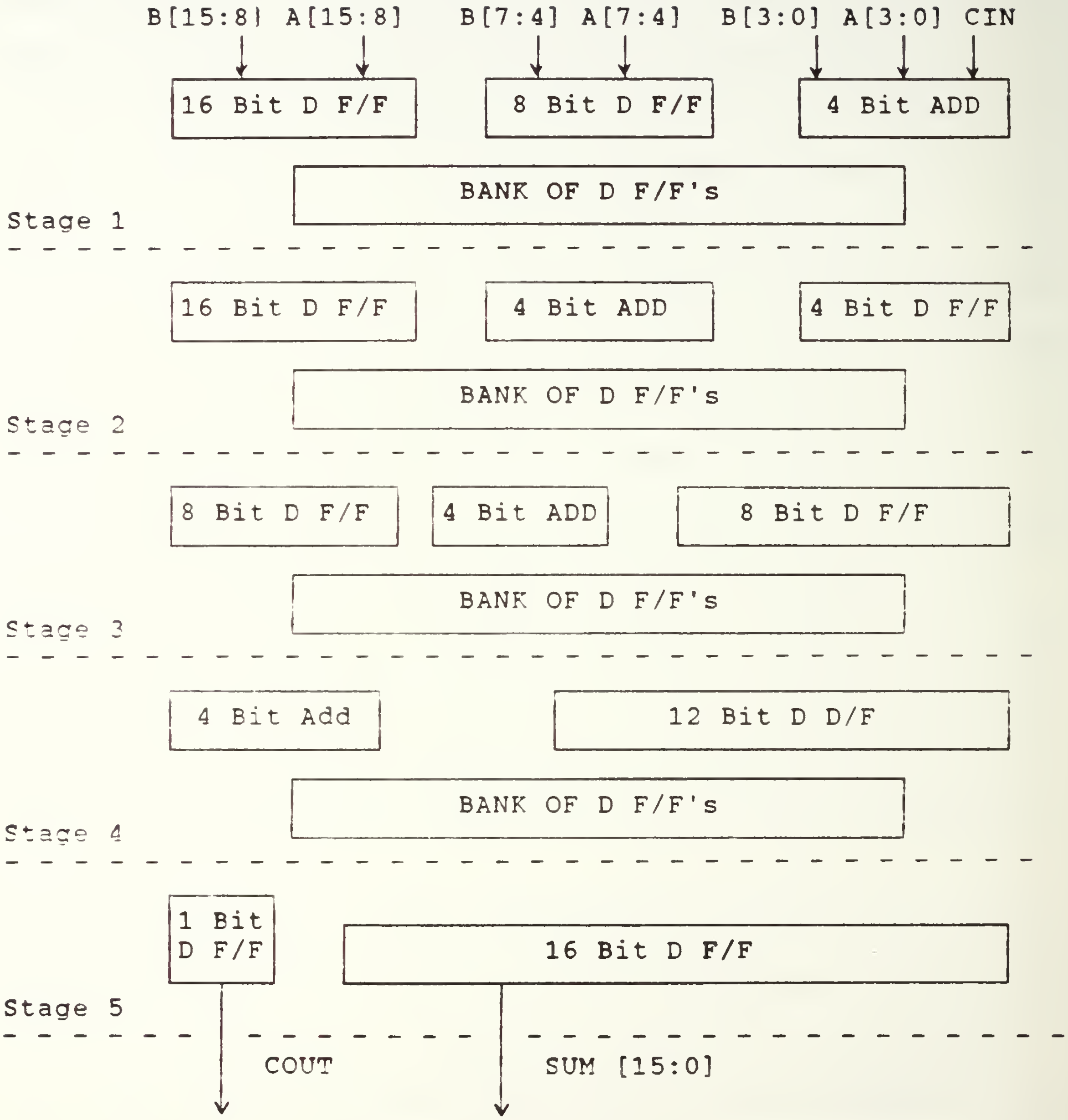


Figure 3.13 16 Bit Pipelined Adder Design

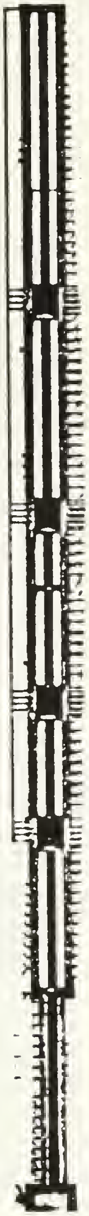


Figure 3.14 List-Best Floorplanned 16 Bit Pipelined Adder

As a size and routing comparison, Figure 3.15 shows the same adder modules stages placed manually. The size of this module was 55.41 X 91.42 mils, resulting in a total area of 5066.50 square mils.

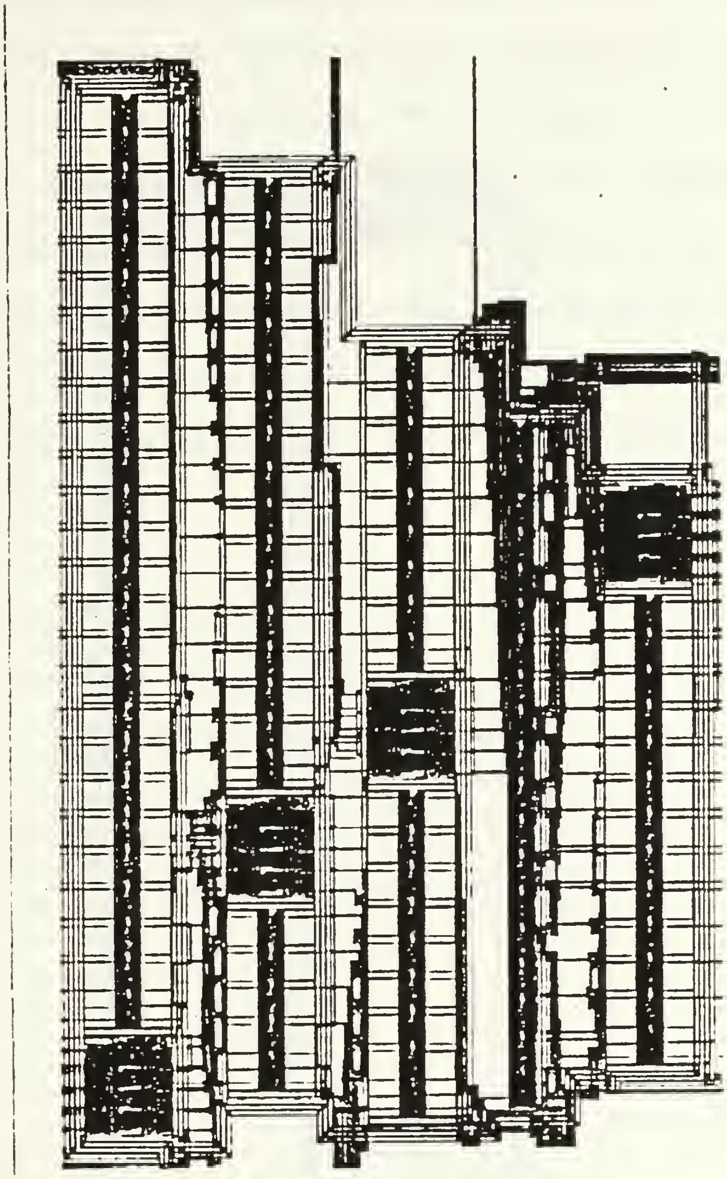


Figure 3.15 Manually Floorplanned
16 Bit Pipelined Adder

Although this design was significantly smaller than the one floorplanned by the system, routing and fusion of the module floorplanned by the system took a significantly shorter time to complete. The system floorplanned module took 10 to 15 minutes to route, while the manually floorplanned module took 1 to 2 hours to complete the routing.

Figure 3.16 shows the pipelined adder with the pads attached. This figure was included to demonstrate the significant increase in chip size experienced with the addition of pads and associated routing. The size of the chip increased to 171.83 X 385.83 mils, resulting in a total area of 66,297.17 square mils.

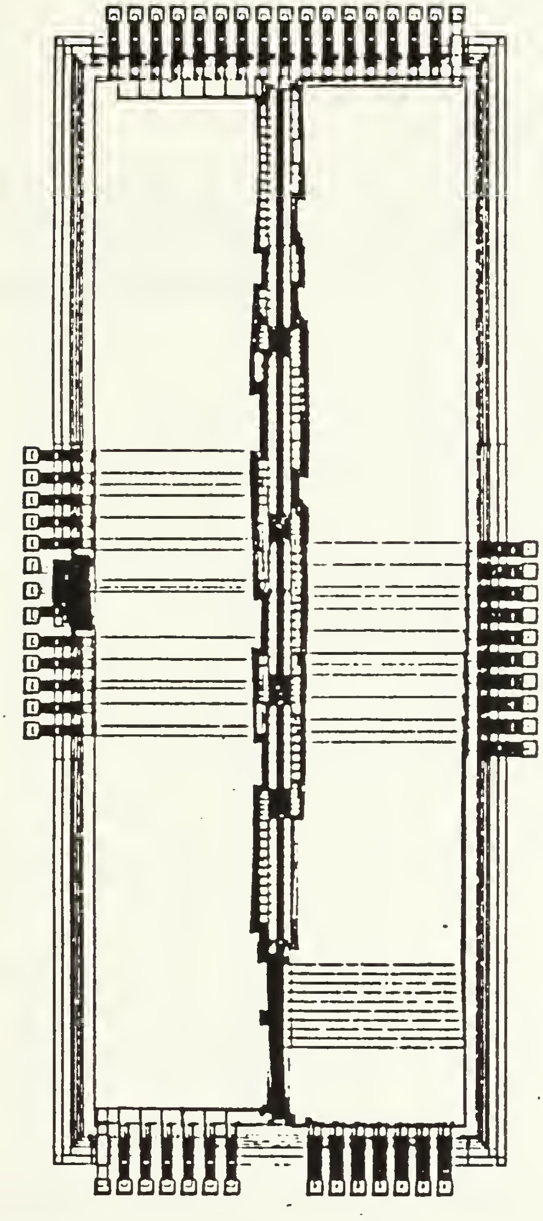


Figure 3.16 16 Bit Pipelined Adder With Pads

Simulation was performed for logic validation using the Functional Simulator. Various combinations of binary integers were placed on the input signal buses, the system clocks cycled, and test results were observed on the output signal buses. Figure 3.17 shows a sample of a simulation test run extracted from the Functional Simulation output form.

```
> is of type module with 48 ports
> port 1 I TRUE to NC = H
> port 3 I FALSE to NC = L
> port 5 O a1[15:0] to NC*16 = 1111111111111110
> port 7 I a[15:0] to NC*16 = HHHHHHHHHHHHHHHH
> port 9 O b1[15:4] to NC*12 = 111111111111
> port 11 I b[15:0] to NC*16 = HHHHHHHHHHHHHHHH
> port 13 O clo to NC = 1
> port 15 CI phase_a to NC = 1
> port 17 CI phase_b to NC = 0
> port 20 O a2[15:8] to NC*8 = 11111111
> port 21 O a2[3:0] to NC*4 = 1111
> port 23 O b2[18:8] to NC*8 = 11111111
> port 25 O c2o to NC = 1
> port 27 O d0[3:0] to NC*4 = 1110
> port 30 O a3[15:12] to NC*4 = 1111
> port 31 O a3[3:0] to NC*4 = 1111
> port 33 O b3[15:12] to NC*4 = 1111
> port 35 O c3o to NC = 1
> port 37 O d3[7:0] to NC*8 = 11111110
> port 39 O a4[3:0] to NC*4 = 1111
> port 41 O c4 to NC = 1
> port 43 O d4[11:0] to NC*12 = 111111111110
> port 45 O carry to NC = 1
> port 47 O sum[15:0] to NC*16 = 1111111111111110
```

Figure 3.17 Simulation Results of 16 Bit Pipelined Adder

Timing analysis was performed to investigate the worst case delay in each stage of the adder. Figure 3.18 through Figure 3.22 list the worst case propagation delays and identifies the worst case signal for each of the five stages of the pipelined adder. The data indicate that the largest propagation delay was 9.5 ns which occurred in stages 1 through 4. This delay is attributed to the library 4 bit adder used in each of these stages.

OUTPUT DELAYS (ns)

Output	Ph1(r) Min	Delay Max	Ph2(r) Min	Delay Max
a1[0]	3.5	4.4	3.5	4.4
a1[10]	4.3	4.5	---	---
a1[11]	4.3	4.5	---	---
a1[12]	4.3	4.5	---	---
a1[13]	4.3	4.5	---	---
a1[14]	4.3	4.5	---	---
a1[15]	4.3	4.5	---	---
a1[1]	3.1	6.3	3.1	6.3
a1[2]	2.3	8.0	2.3	8.0
a1[3]	3.1	9.2	3.1	9.2
a1[4]	3.8	4.0	---	---
a1[5]	3.8	4.0	---	---
a1[6]	3.8	4.0	---	---
a1[7]	3.8	4.0	---	---
a1[8]	4.3	4.5	---	---
a1[9]	4.3	4.5	---	---
b1[10]	4.3	4.5	---	---
b1[11]	4.3	4.5	---	---
b1[12]	4.3	4.5	---	---
b1[13]	4.3	4.5	---	---
b1[14]	4.3	4.5	---	---
b1[15]	4.3	4.5	---	---
b1[4]	3.8	4.0	---	---
b1[5]	3.8	4.0	---	---
b1[5]	3.0	4.0	---	---
b1[6]	3.8	4.0	---	---
b1[7]	3.8	4.0	---	---
b1[8]	4.3	4.5	---	---
b1[9]	4.3	4.5	---	---
clo	1.9	9.5	1.9	9.5

Figure 3.18 Stage_1 Output Delays

OUTPUT DELAYS (ns)

Output	Ph1(r)	Delay	Ph2(r)	Delay
	Min	Max	Min	Max
a2[0]	1.6	5.2	1.6	5.2
a2[10]	4.3	4.5	---	---
a2[11]	4.3	4.5	---	---
a2[11]	4.3	4.5	---	---
a2[12]	4.3	4.5	---	---
a2[13]	4.3	4.5	---	---
a2[14]	4.3	4.5	---	---
a2[15]	4.3	4.5	---	---
a2[1]	3.1	6.3	3.1	6.3
a2[2]	2.3	8.0	2.3	8.0
a2[3]	3.1	9.2	3.1	9.2
a2[8]	4.3	4.5	---	---
a2[9]	4.3	4.5	---	---
b2[10]	4.3	4.5	---	---
b2[11]	4.3	4.5	---	---
b2[12]	4.3	4.5	---	---
b2[13]	4.3	4.5	---	---
b2[14]	4.3	4.5	---	---
b2[15]	4.3	4.5	---	---
b2[8]	4.3	4.5	---	---
b2[9]	4.3	4.5	---	---
c2o	1.9	9.5	1.9	9.5
d0[0]	3.5	3.7	---	---
d0[1]	3.5	3.7	---	---
d0[2]	3.5	3.7	---	---
d0[3]	3.5	3.7	---	---

Figure 3.19 Stage_2 Output Delays

OUTPUT DELAYS (ns)

Output	Ph1(r)	Delay	Ph2(r)	Delay
	Min	Max	Min	Max
a3[0]	1.6	5.2	1.6	5.2
a3[12]	3.8	4.0	---	---
a3[13]	3.8	4.0	---	---
a3[14]	3.8	4.0	---	---
a3[15]	3.8	4.0	---	---
a3[1]	3.1	6.3	3.1	6.3
a3[2]	2.3	8.0	2.3	8.0
a3[3]	3.1	9.2	3.1	9.2
b3[12]	3.8	4.0	---	---
b3[13]	3.8	4.0	---	---
b3[14]	3.8	4.0	---	---
b3[15]	3.8	4.0	---	---
c3o	1.9	9.5	1.9	9.5
d3[0]	3.8	4.0	---	---
d3[1]	3.8	4.0	---	---
d3[2]	3.8	4.0	---	---
d3[3]	3.8	4.0	---	---
d3[4]	3.8	4.0	---	---
d3[5]	3.8	4.0	---	---
d3[6]	3.8	4.0	---	---
d3[7]	3.8	4.0	---	---

Figure 3.20 Stage_3 Output Delays

```
                    OUTPUT DELAYS (ns)
Output           Ph1(r)  Delay       Ph2(r)  Delay
                 Min       Max       Min       Max
a4[0]            1.6       5.2       1.6       5.2
a4[1]            3.1       6.3       3.1       6.3
a4[2]            2.3       8.0       2.3       8.0
a4[3]            3.1       9.2       3.1       9.2
c4               1.9       9.5       1.9       9.5
d4[0]            4.0       4.2       ---       ---
d4[10]           4.0       4.2       ---       ---
d4[11]           4.0       4.2       ---       ---
d4[1]            4.0       4.2       ---       ---
d4[1]            4.0       4.2       ---       ---
d4[2]            4.0       4.2       ---       ---
d4[3]            4.0       4.2       ---       ---
d4[4]            4.0       4.2       ---       ---
d4[5]            4.0       4.2       ---       ---
d4[6]            4.0       4.2       ---       ---
d4[7]            4.0       4.2       ---       ---
d4[8]            4.0       4.2       ---       ---
d4[9]            4.0       4.2       ---       ---
```

Figure 3.21 Stage_4 Output Delays

```
                OUTPUT DELAYS (ns)
Output           Ph1(r)  Delay        Ph2(r)  Delay
                 Min       Max        Min       Max
carry            3.3       3.5        ---       ---
sum[0]           4.3       4.5        ---       ---
sum[10]          4.3       4.5        ---       ---
sum[11]          4.3       4.5        ---       ---
sum[12]          4.3       4.5        ---       ---
sum[13]          4.3       4.5        ---       ---
sum[14]          4.3       4.5        ---       ---
sum[15]          4.3       4.5        ---       ---
sum[1]           4.3       4.5        ---       ---
sum[2]           4.3       4.5        ---       ---
sum[3]           4.3       4.5        ---       ---
sum[4]           4.3       4.5        ---       ---
sum[5]           4.3       4.5        ---       ---
sum[6]           4.3       4.5        ---       ---
sum[7]           4.3       4.5        ---       ---
sum[8]           4.3       4.5        ---       ---
sum[9]           4.3       4.5        ---       ---
```

Figure 3.22 Stage_5 Output Delays

4. Results

The results of the comparison between a standard 16 bit library adder and custom pipelined 16 bit adder are shown in Table VIII. These figures do not include delay associated with interstage flip-flops.

TABLE VIII
SIZE AND PERFORMANCE SUMMARY

	TOTAL AREA (Sq mils)	MAXIMUM PROPAGATION Delay (ns)
GENESIL 16 BIT ADDER	219.33	27.2
CUSTOM 16 BIT PIPELINED ADDER	7424.87	9.5

E. PERFORMANCE SUMMARY

Table IX is a summary of the size and performance results of the adders designed and constructed in this chapter.

TABLE IX
ADDERS SIZE AND PERFORMANCE SUMMARY

	TOTAL AREA (Sq mils)	MAXIMUM PROPAGATION Delay (ns)
GENESIL FULL ADDER	26.15	5.2
CUSTOM FULL ADDER	18.34	5.0
GENESIL 4 BIT ADDER	65.07	9.4
CUSTOM 4 BIT CLA ADDER	107.76	9.0
GENESIL 16 BIT ADDER	219.33	27.2
CUSTOM 16 BIT PIPELINED ADDER	7424.87	9.5

The data indicate that there is no significant performance advantage gained with the custom full and 4 bit adders. The custom 16 bit pipelined adder data clearly illustrate the performance advantage gained by pipelining.

IV. MULTIPLIER CIRCUITS

A. INTRODUCTION.

The binary multiplier is a major component of signal processor filters. Conventional ALU add and shift multiply functions and parallel multiplier circuits are not adequate for the speed requirements of the high speed processor. This chapter presents performance data on 4, 8, and 16 bit unsigned library multipliers, which are compared to a custom 4 and 16 bit pipelined multiplier.

The multiplication add-and-shift algorithm for two n-bit binary numbers are represented by Equation 4.1 [Refs. 11 and 16].

$$\underline{p} = \sum_{k=0}^{n-1} 2^{k} a_{k} \underline{b} \qquad (4.1)$$

> $\underline{b}$ represents the n-bit multiplication vector, $\underline{a}$ represents bits n of the multiplier vector $\underline{a}$ and $\underline{p}$ represents the 2n bit product vector.[Ref. 16:p. 11]

This concept is illustrated in Figure 4.1 [Ref. 16:p. 11] for the product of two 8-bit integers. The multiplication of the two 8-bit integers results in eight partial products, generated from the ANDING of the multiplicand (MC) and multiplier (MP) bits, which are then added to form the final product.

```
                        X7 X6 X5 X4 X3 X2 X1 X0—MULTIPLICAND
                        Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0—MULTIPLIER
                        -----------------------
                        A7 A6 A5 A4 A3 A2 A1 A0—A PARTIAL
                     B7 B6 B5 B4 B3 B2 B1 B0        PRODUCT
                  C7 C6 C5 C4 C3 C2 C1 C0
               D7 D6 D5 D4 D3 D2 D1 D0
            E7 E6 E5 E4 E3 E2 E1 E0
         F7 F6 F5 F4 F3 F2 F1 F0
      G7 G6 G5 G4 G3 G2 G1 G0
   H7 H6 H5 H4 H3 H2 H1 H0
-------------------------------------------------------------
S15S14S13S12S11S10S9 S8 S7 S6 S5 S4 S3 S2 S1 S0—FINAL
                                                 PRODUCT
```

Figure 4.1 Two 8-Bit Integer Product

B. 4, 8, AND 16 BIT GENESIL LIBRARY MULTIPLIER

1. Multiplier Block Array Core

The Genesil library multiplier block array core is an array of half and full adders which provides a parallel multiplier scheme for integer and fraction/mantissa portions of floating-point numbers. In depth parallel multiplier theory and schematics are contained in Reference 4:pp. 344-348. The multiplier block array core and operation illustrating the product of 111001 multiplied by 1101 is shown in Figure 4.2 [Ref. 13:p. 4.3]. The block is designed for unsigned integer multiplication and requires external circuitry for signed operations. The least significant (LS) bits are produced directly from the array, but an external adder is required to complete the partial product addition of the most significant (MS) bits. The multiplier and multiplicand widths can be varied from 4 to 32 bits, but the multiplier width cannot exceed the multiplicand width.

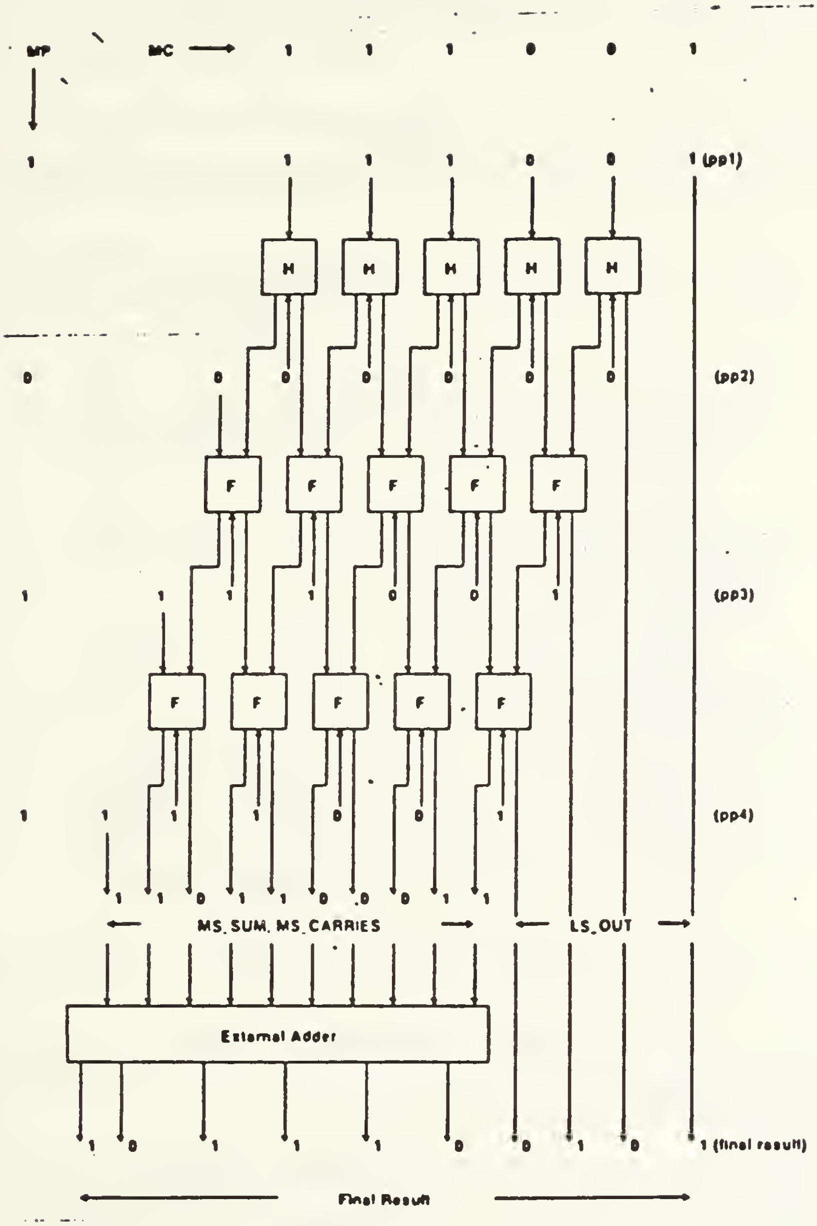


Figure 4.2 Multiplier Block Array Core and External Adder

2. 4, 8, and 16 Bit Multiplier Data

Three modules, containing 4, 8, and 16 bit Genesil library multipliers and external adders, were constructed from the Specification Menu. Each was simulated for correct logic and processed by the Timing Analyser for propagation delay data. The results are summarized in Table X.

TABLE X
4, 8, AND 16 BIT OUTPUT PROPAGATION DELAYS

	OUTPUT PROPAGATION DELAYS (ns)	
	MIN	MAX
4 BIT		
MS_SUM	3.2	10.9
LS_OUT	3.1	10.3
MS_OUT	5.3	18.8
8 BIT		
MS_SUM	4.1	24.4
LS_OUT	3.2	23.4
MS_OUT	5.9	38.2
16 BIT		
MS_SUM	4.6	51.0
LS_OUT	3.2	49.4
MS_OUT	6.9	77.0

The MS_SUM data are the propagation delays of the array core only. MS_OUT is the total propagation delay of both the core and external adder.

The addition of D flip/flops between the core and external adder in Figure 4.2 decreased the propagation delay driving the maximum clock speed allowable for the circuit to that of the MS_SUM output propagation delay. The D F/F inputs were MS_SUM, MS_CARRIES, and LS_OUT. MS_SUM and MS_CARRIES were then clocked into the external adder or next stage of the pipeline. Table XI illustrates the theoretical allowable clock speed of a circuit using the multiplier modules considering each module with and without the D flip/flop insertion. The modules without D flip/flops inserted between the core and external adder, clock speeds were calculated assuming there was a D flip/flop attached to the outputs of the external adder and LS_OUT.

TABLE XI
THEORETICAL CLOCK SPEED OF
4, 8, AND 16 BIT LIBRARY MULTIPLIER

	CLOCK SPEED (MHZ)	
	WITHOUT D F/F INSERTED	WITH D F/F INSERTED
4 BIT MULTIPLIER	39.5	57.4
8 BIT MULTIPLIER	22.3	32.3
16 BIT MULTIPLIER	11.3	17.3

The data indicate that there was a significant increase of the allowable clock speed of a circuit using the multiplier modules with the addition of the D flip/flop inserted between the core and external adder.

C. 4 BIT PIPELINED MULTIPLIER

1. Introduction

This section presents performance data for a 4 bit pipelined multiplier using the Wallace Tree structure. Figure 4.3 illustrates a 4X4 multiplication in dot form [Ref. 16].

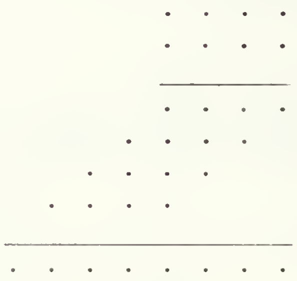


Figure 4.3 4X4 Multiply

After the partial products are formed, the three right columns of partial products are shifted down to form a pyramid or tree, as illustrated in Figure 4.5 [Ref. 16].

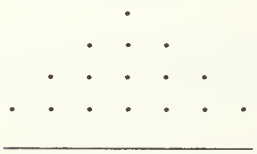


Figure 4.4 Wallace Tree Partial Products

Next, 3-input, 2-output full adders are used to compute carry save addition (CSA) for column reduction.

> To reduce these columns of height h, CSA is used to reduce three dots of column height to two. These two output dots, which represent the familiar sum and carry outputs of a full adder, are placed in the next level of the tree structure in their appropriate positions.[Ref. 16:p.16]

This concept is illustrated in Figure 4.5 for a 4X4 multiplication.

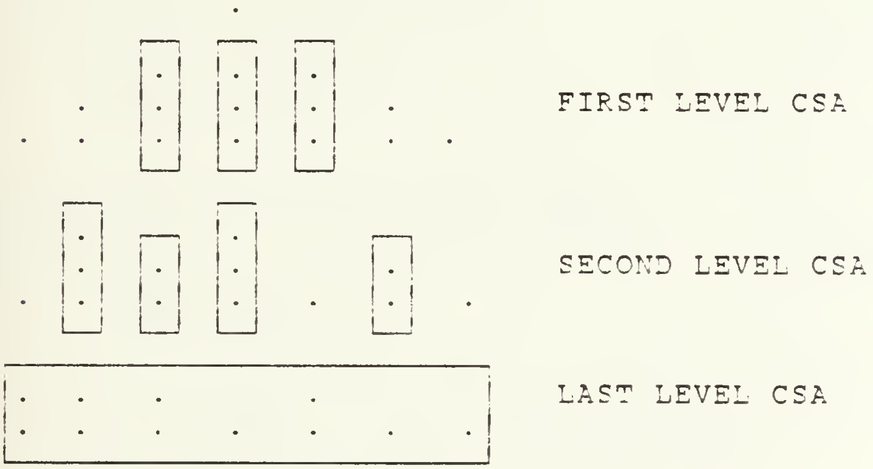


Figure 4.5 CSA Reduction 4X4 Multiplication

Once reduced to the last level addition, various CLA and pipelined ripple adder designs are available for increased performance.

2. 4 Bit Pipelined Multiplier Design

The 4 bit pipelined multiplier was designed using the Wallace Tree Structure, with D flip/flops inserted for pipelining. A block diagram of the module is shown in Figure 4.6. All partial products were generated simultaneously by the 16 AND gates. Next, the partial products were reduced using the Wallace Tree concept described in section 1. The final level additions were computed by a pipelined Genesil library 2 bit ripple adder and a 3 bit ripple adder.

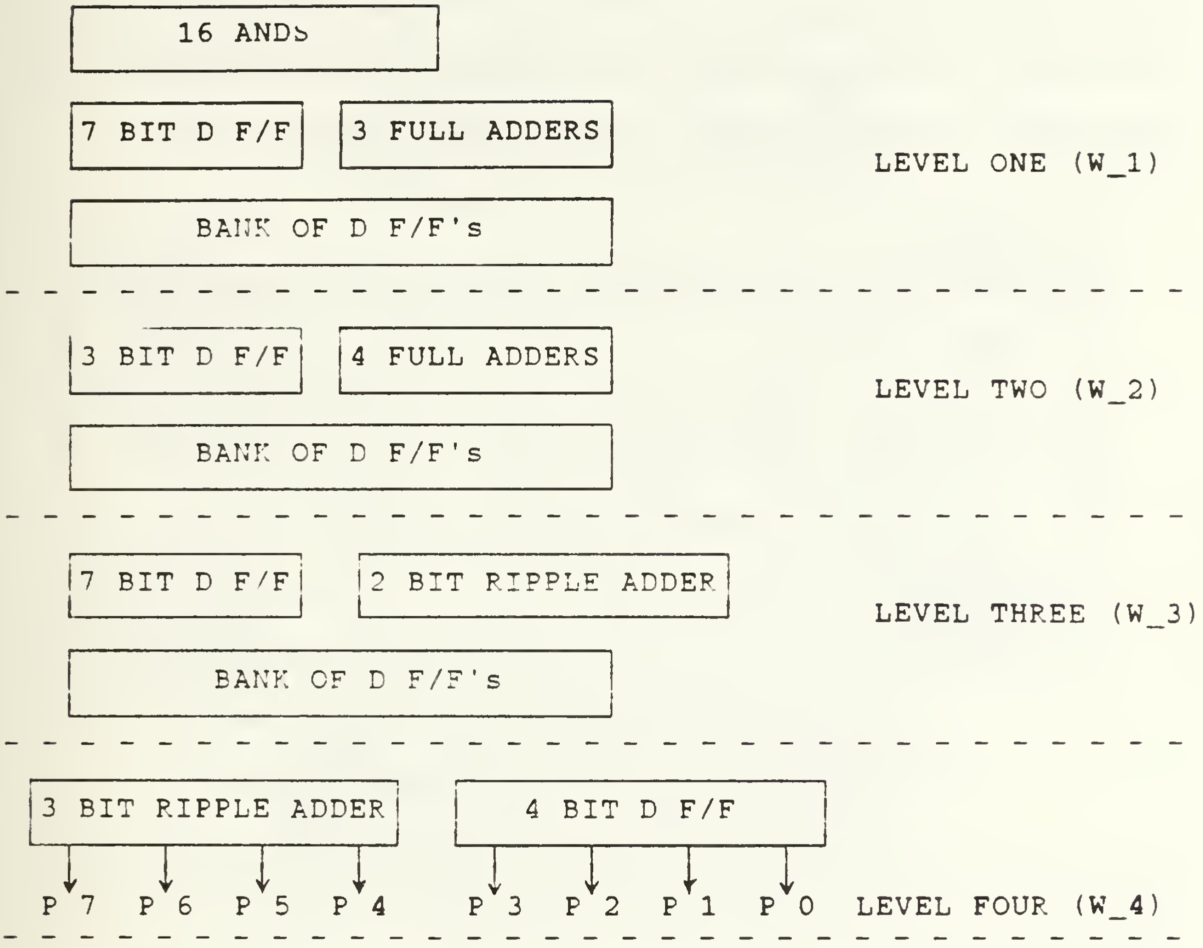


Figure 4.6 Custom 4 Bit Pipelined Multiplier Block Diagram

The module consisted of 4 blocks called W_1, W_2, W_3, and W_4. The module was attached to a chip, floorplanned, and simulated to test for correct logic. Figure 4.7 depicts the floorplan.

Figure 4.8 shows the 4 Bit Multiplier Chip with pads, clock, ground, and power. Timing analysis was performed by the system Timing Analyser for output propagation delays at each level. The results are presented in Table XII.

TABLE XII
4 BIT PIPELINED MULTIPLIER OUTPUT PROPAGATION DELAYS

BLOCK	OUTPUT PROPAGATION DELAYS (ns)	
	MIN	MAX
W_1	3.7	6.8
W_2	3.5	4.7
W_3	4.3	6.6
W_4	3.5	7.6

The data indicate that the longest output propagation delay was 7.6 ns, which occurred in block W_4.

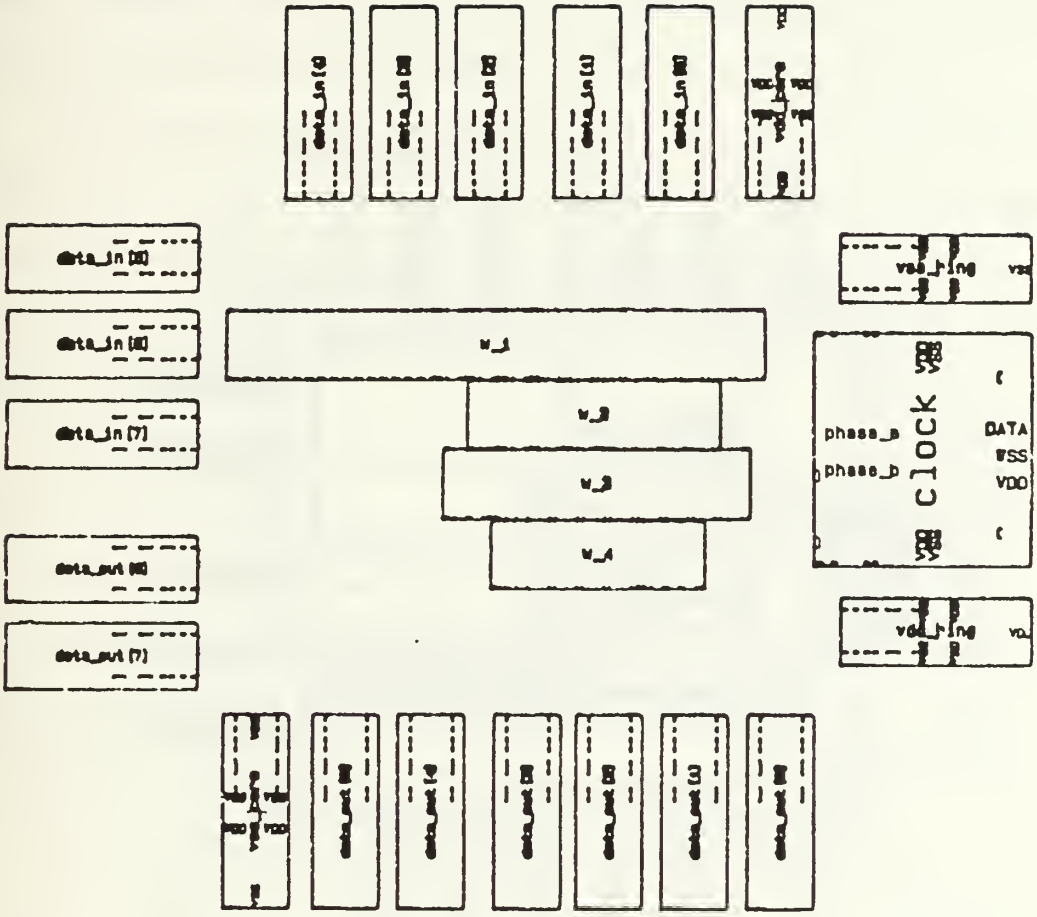


Figure 4.7 4 Bit Pipelined Multiplier Floorplan

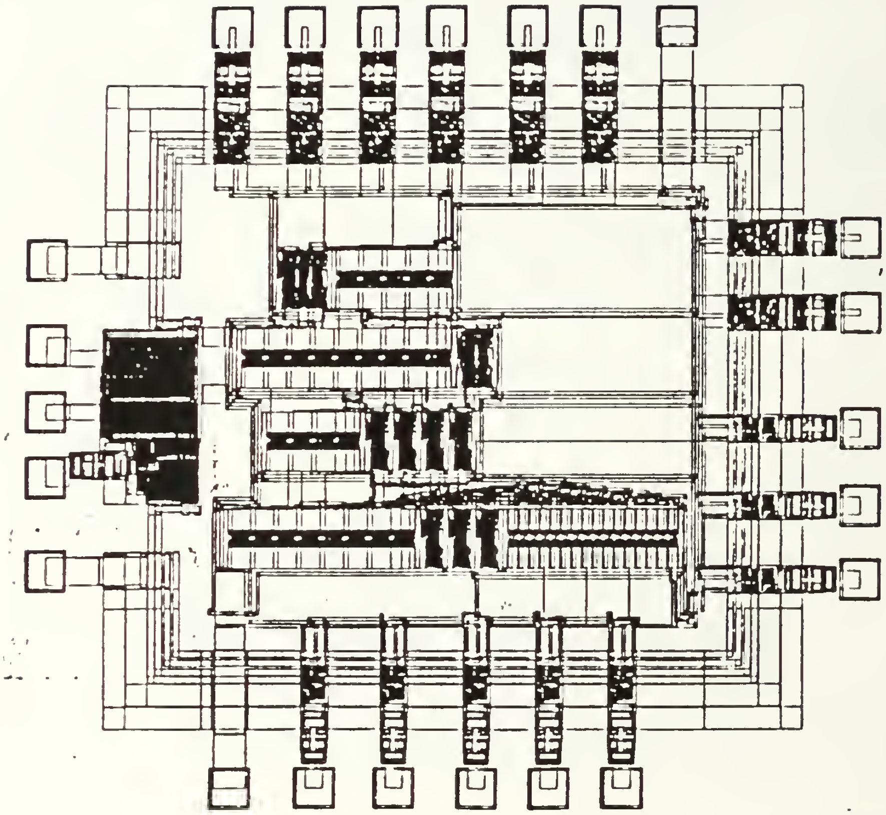


Figure 4.8 4 Bit Pipelined Multiplier Chip

D. 16 BIT PIPELINED MULTIPLIER

The purpose of the work reported in this section was to build a strictly high performance 16 bit pipelined multiplier chip which could be rapidly tested and de-bugged. Two designs were considered. They were the Wallace Tree structure and ripple adder design using a pipelined parallel multiplier with all partial products computed prior to array entry.

The Wallace Tree structure was rejected because, while it saved only two levels of logic, the design presented serious de-bugging difficulties. It was found to be extremely difficult to trace and debug signal errors when the column height was 16.

The design used was the pipelined parallel multiplier. The primary advantage of this design was found to be the relative ease of de-bugging the chip. The primary disadvantage was the additional cost in hardware and chip size associated with D flip/flop delays used to align and save intermediate results [Ref. 12:pp. 51-53].

1. 16 Bit Pipelined Multiplier Design

The 16 bit pipelined multiplier was designed using a pipelined parallel multiplier, and pipelined ripple carry adder hardware for summing the final partial products. A design block diagram is shown in Figure 4.9. All partial products were generated simultaneously by the 256 AND gates. This initial partial product generation is also necessary for the Wallace Tree structure. Partial product reduction can be accomplished by Booth's modified algorithm and read only memories (ROM) [Ref. 19], but neither were pursued in this design. The partial products were then reduced with full adders, aligned, and rippled through the array with D flip/flops.

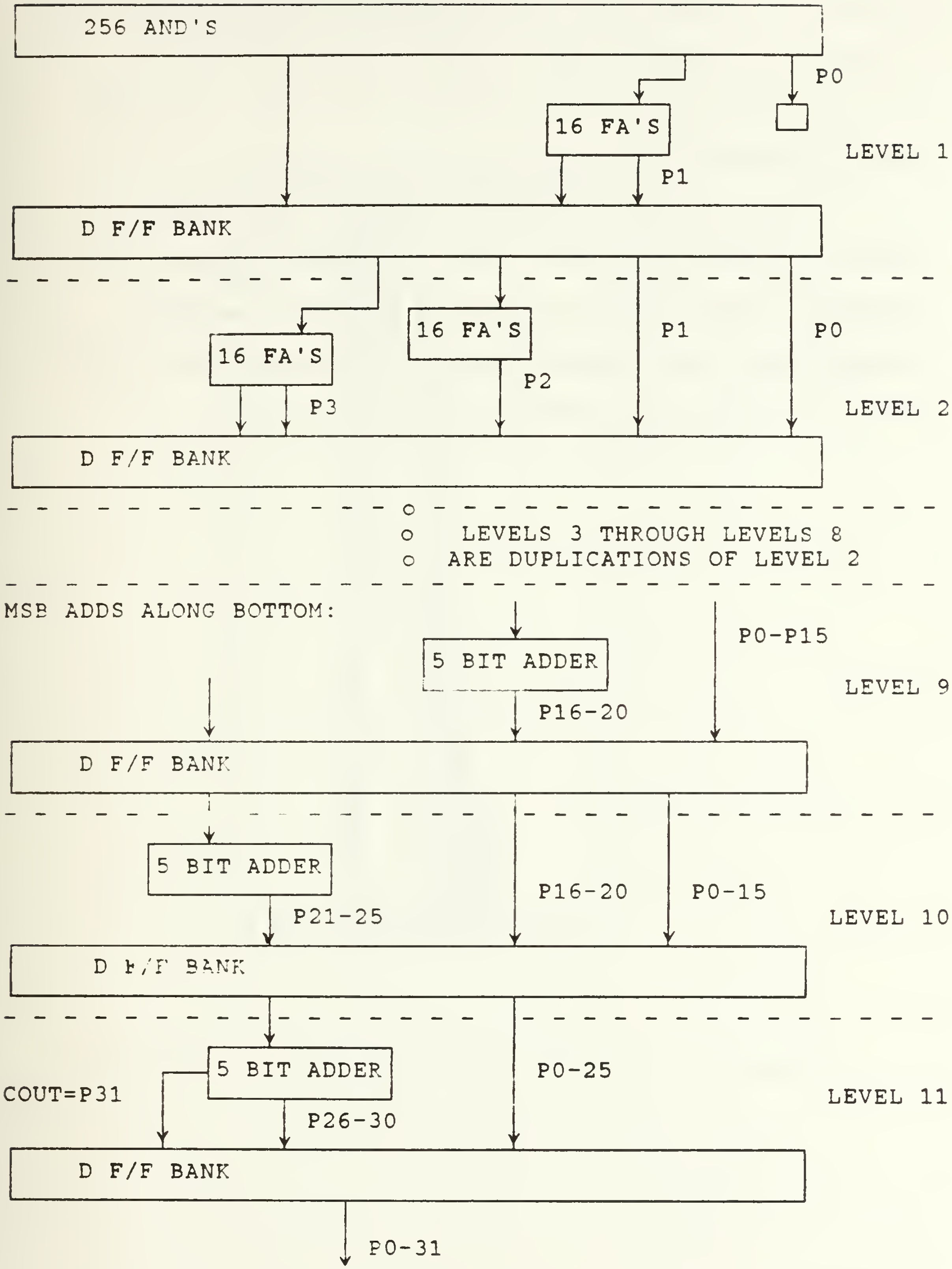


Figure 4.9 Custom 16 Bit Pipelined Multiplier Block Design

Eleven levels of blocks were used to construct the multiplier. These were then attached to a chip, which also included input/output pads, clock, power, and ground. The chip was then floorplanned, and the floorplan, with pads, is shown in Figure 4.10.

The chip was tested for correct logic using the system simulation feature. Ten to fifteen random 16 bit unsigned integers were inserted on the input signals. Although the tests run were not all inclusive, the results indicated correct logic for the inputs tested.

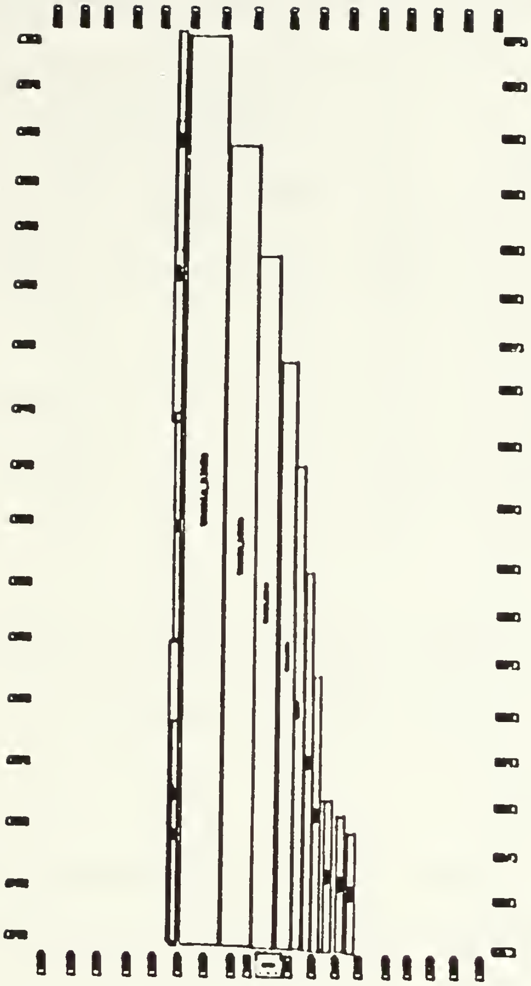


Figure 4.10 16 Bit Pipelined Multiplier Floorplan

Timing analysis was performed by the system Timing Analyser for output propagation delays at each level. The results are presented in Table XIII.

TABLE XIII
16 BIT PIPELINED MULTIPLIER OUTPUT PROPAGATION DELAYS

LEVEL	OUTPUT PROPAGATION DELAYS (ns)	
	MIN	MAX
1	4.7	5.8
2-8	7.2	10.4
9	7.9	8.1
10	7.9	8.1
11	7.9	8.1

The data indicate that the longest delay in the circuit is 10.4 ns, occurring in each level 2 through 8.

E. PERFORMANCE RESULTS

Table XIV is a summary of the performance results of the multipliers designed and constructed in this chapter.

TABLE XIV
MULTIPLIER PERFORMANCE RESULTS

	MAX DELAY (ns)	NO. OF STAGES	CLOCK RATE (MHZ)
4 BIT GENESIL (WITH LATCH)	10.9	1	57.4
8 BIT GENESIL (WITH LATCH)	24.4	1	32.3
16 BIT GENESIL (WITH LATCH)	51.0	1	17.3
4 BIT WALLACE (PIPELINED)	7.6	1	70.9
16 BIT PARALLEL (PIPELINED)	10.4	8	59.1

The data clearly illustrate the performance advantage gained by using the custom pipelined multipliers for high performance tasks.

V. CONCLUSIONS

A. SUMMARY

This thesis has described the applications of silicon compilers, and the design methodology of the Genesil Silicon Compiler. The Genesil Silicon Compiler methodology was demonstrated with the design and verification of custom pipelined adder and multipler circuits.

The Genesil Silicon Compiler system is a rapid and efficient stand-alone tool for algorithm to hardware implementation and verification. Rapid iterative design, simulation, and timing analysis is possible because the system requires no user initiated programming.

The Genesil system user's manuals state that it is assumed the user has attended the Genesil Silicon Compiler user school. The manuals are reference manuals, and not tutorials for new users. The new user, however, can rapidly learn the system.

The user should thoroughly pre-plan design and performance specifications because there is not a plot "screen dump" capability on the system. The user must manually track and record all signal and object changes if an updated design plot is desired at the end of a session.

Object compiling and channel routing times for the circuits designed in this thesis were longer than anticipated. In order to expedite object compiling during design iterations and de-bugging, object sizes (i.e., blocks, modules) should be as small as practicable. The system auto placement and routing features decreased routing times, but were less efficient than manual placement for overall object size.

Complete chips, with all associated hardware, consumed much system memory. During thesis research, chips and objects were stored on tape backups when memory availability became critical. All design and performance specifications can be verified at the block and module levels, which saves memory and routing time.

B. RECOMMENDATIONS

The following recommendations should be considered:

1. Research the area of optimum chip test algorithms prior to foundary tapeout. Investigate the full Genesil Compiler System Corporation's capabilities in the test area.

2. Purchase a plotter for plot "screen dumps" for rapid intermediate design schematics.

3. Transfer the system to the VAX 785 for more memory capability and faster tape storage capabilities.

4. Following system transfer to the VAX 785, establish a user custom library for high performance modules including pipelined integer multipliers, floating point multipliers, signed multipliers, and adders.

5. Do design, layout, simulation and Timing Analysis without pads for memory and routing time efficiency.

APPENDIX

GENESIL SILICON COMPILER TUTORIAL

A. INTRODUCTION

The purpose of this tutorial is to guide the new user through the mechanics of a Genesil system hierarchical top-down chip design. Designs may be implemented either top-down or bottom up. The tutorial begins with designing two basic blocks, followed by a multiplier module, and summarized with the design of a chip which uses the two blocks as its core.

Prior to beginning the initial session, the user should become familiar with the System Description Users Manual, in particular, Chapters 2 and 3. The next manual of interest is the System Description Application Commands manual, which contains detailed explanations of user invoked commands. The new user should periodically refer to Appendix A (Genesil System Menu Map) of the System Description Application Commands manual during initial sessions.

1. Design Method

All design and performance specifications should be pre-planned, including a detailed sketch with all signal names. The basic stand-alone object which can be attached to a chip is the block. Blocks may be attached to modules or chips, but not to other blocks. Modules, the intermediate object in the hierarchy, may be attached to other modules and chips. There are no chip size constraints in the Genesil system, although this is dicated by the selected foundary. Large designs can, therefore, be implemented with chipsets.

2. Operating System

Genesil runs under the UNIX operating system. The user is referred to the UNIX For Genesil Users manual for detailed UNIX pathnames information. The pathname is the full name of an object in the Genesil system. The user is referred to page 4.8 in the System Description Users Manual for naming conventions details.

B. TUTORBLK_1 BLOCK

This section contains a step-by-step design of a block named tutorblk_1. The block will contain two random logic objects, which are a 4 bit adder (A0), and 5 bit D F/F (DFF1). The pre-planned schematic of the block, including all signal names is shown in Figure A.1.

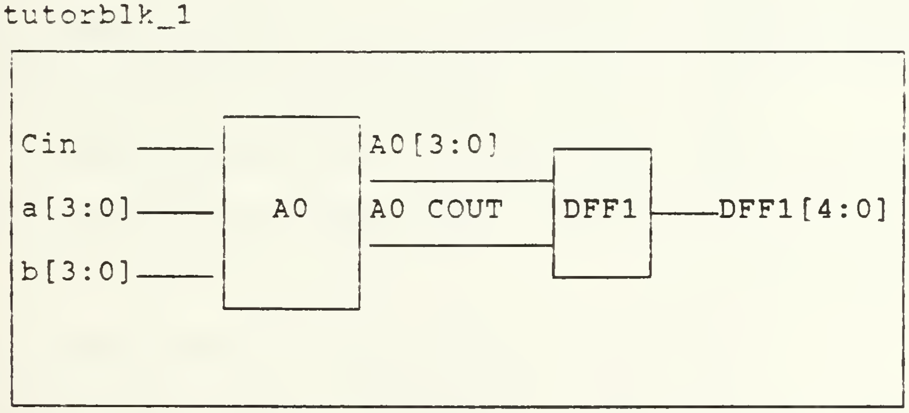


Figure A.1 Tutorblk_1

Since the system is menu driven, it is important to have a detailed schematic with signal names clearly marked. The same signal name cannot both enter and leave the same object.

All commands may be executed by typing in the command next to the prompt followed by a RETURN, by using the arrow buttons located on the upper right side of the keyboard to scroll through the commands followed by a RETURN, or by using the MOUSE. All following command instructions assume the user is using the MOUSE. The instruction select SOME_THING, means use the mouse to move the cross-hairs to SOME_THING and press the execute button (right hand button) on mouse.

1. While in the Executive menu (upper right corner of screen):

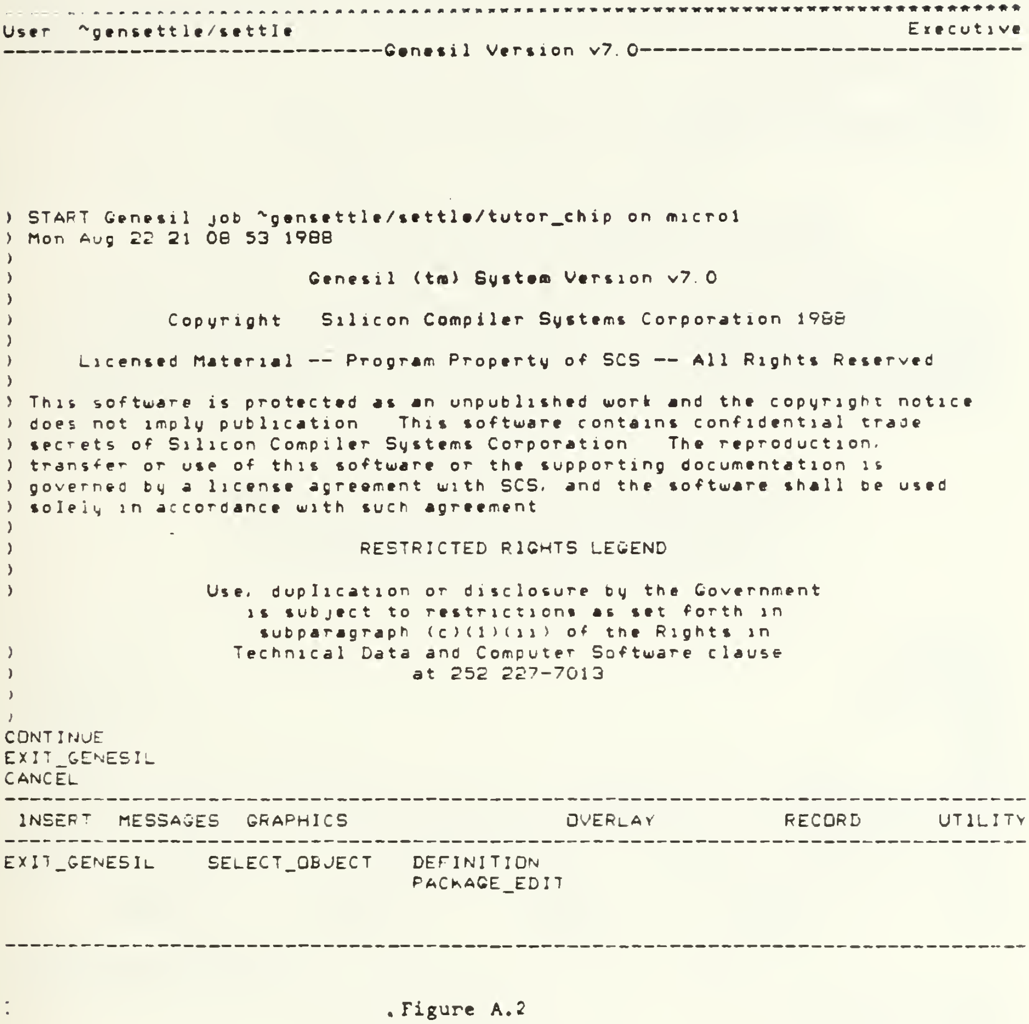
   a. Following LOGIN and GENESIL entry, select CONTINUE.

   b. Select SELECT_OBJECT (Figure A.2). This is normally always the initial command in order to attach objects to the user tree.

   c. Select ATTACH (Figure A.3), followed by NEW (Figure A.4) since this block is the intital object.

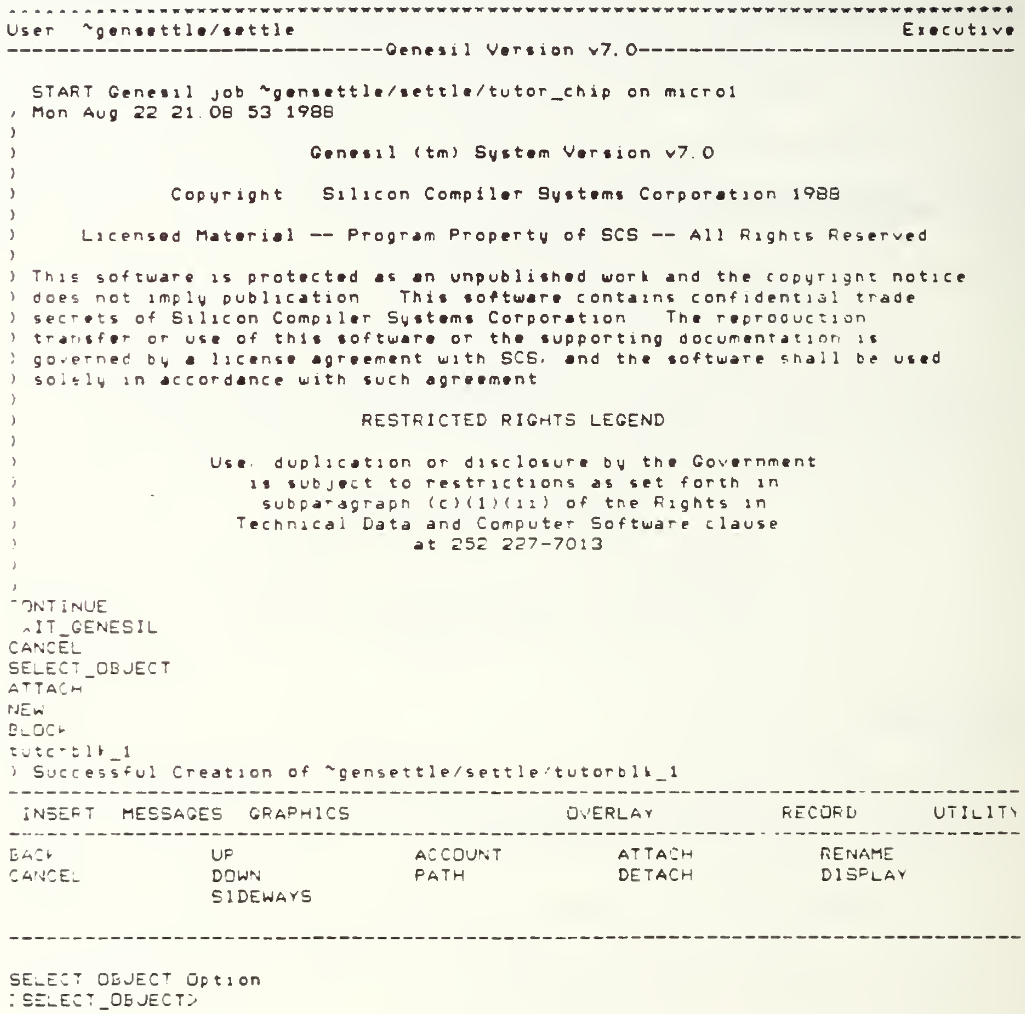
   d. Select BLOCK (Figure A.5) since this is the object type desired.

   e. Next type in tutorblk_1, at the prompt followed by a <CR>. This is now the name of a new, yet to be defined, block, as indicated by the successful creation statement on the screen.



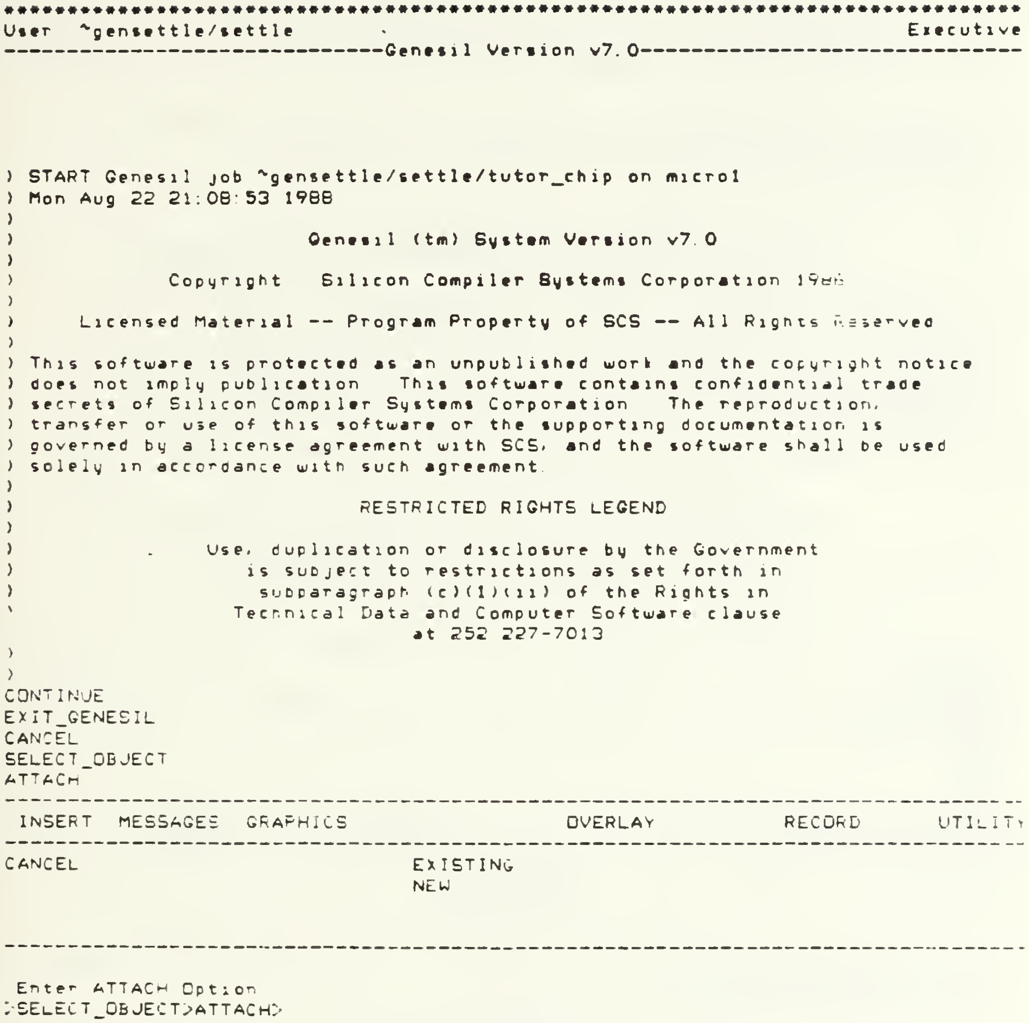
```
--------------------------------------------------------------------------------
User  ~gensettle/settle                                               Executive
------------------------------Genesil Version v7.0------------------------------




) START Genesil job ~gensettle/settle/tutor_chip on micro1
) Mon Aug 22 21 08 53 1988
)
)                         Genesil (tm) System Version v7.0
)
)              Copyright   Silicon Compiler Systems Corporation 1988
)
)     Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication   This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation   The reproduction,
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement
)
)                             RESTRICTED RIGHTS LEGEND
)
)                Use, duplication or disclosure by the Government
                    is subject to restrictions as set forth in
                     subparagraph (c)(1)(ii) of the Rights in
)                  Technical Data and Computer Software clause
)                                 at 252 227-7013
)
)
CONTINUE
EXIT_GENESIL
CANCEL
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
EXIT_GENESIL    SELECT_OBJECT   DEFINITION
                                PACKAGE_EDIT


--------------------------------------------------------------------------------


:
```

Figure A.2



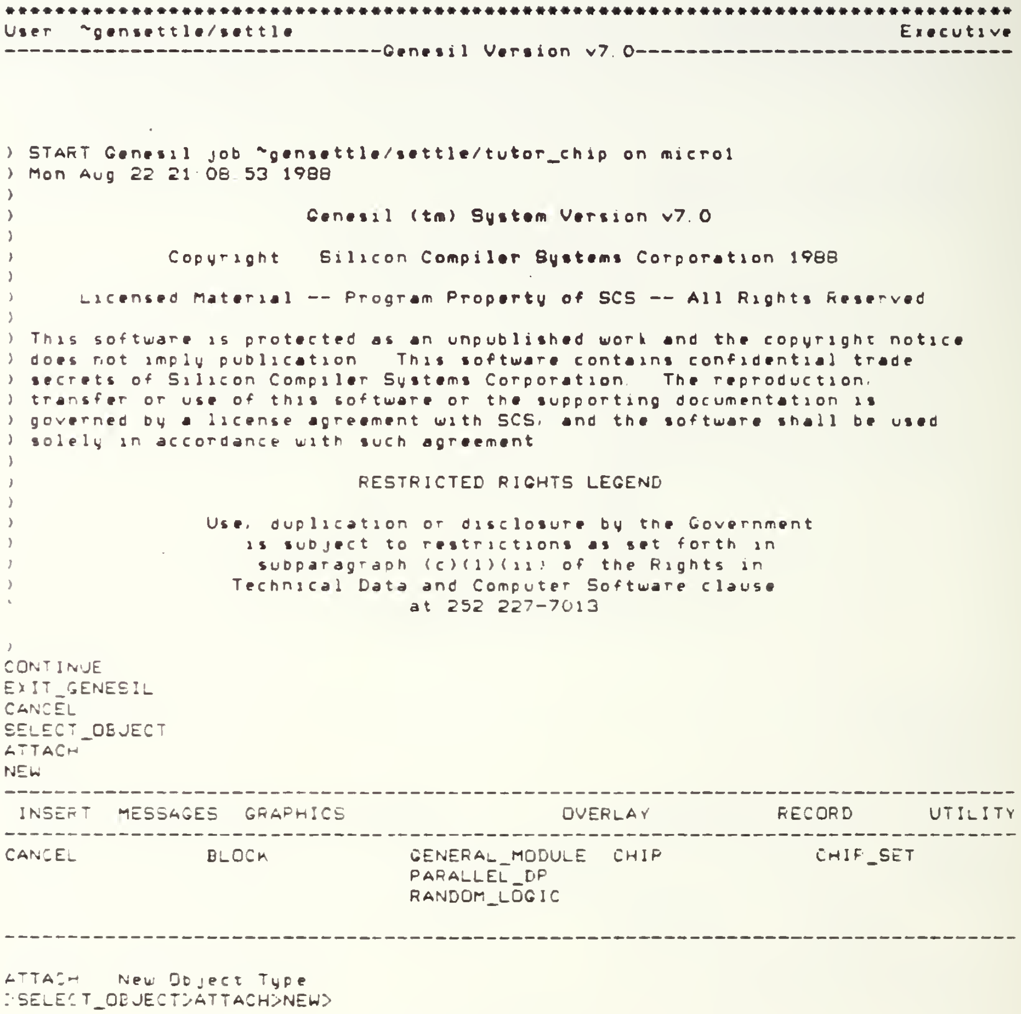
```
********************************************************************************
User  ~gensettle/settle                                                Executive
-----------------------------Genesil Version v7.0------------------------------

  START Genesil job ~gensettle/settle/tutor_chip on micro1
) Mon Aug 22 21.08 53 1988
)
)                        Genesil (tm) System Version v7.0
)
)              Copyright   Silicon Compiler Systems Corporation 1988
)
)     Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication   This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation   The reproduction
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement
)
)                           RESTRICTED RIGHTS LEGEND
)
)                 Use, duplication or disclosure by the Government
)                    is subject to restrictions as set forth in
)                     subparagraph (c)(1)(ii) of the Rights in
)                   Technical Data and Computer Software clause
)                                at 252 227-7013
)
)
CONTINUE
 EXIT_GENESIL
CANCEL
SELECT_OBJECT
ATTACH
NEW
BLOCK
tutorblk_1
) Successful Creation of ~gensettle/settle/tutorblk_1
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
BACK            UP              ACCOUNT         ATTACH          RENAME
CANCEL          DOWN            PATH            DETACH          DISPLAY
                SIDEWAYS

--------------------------------------------------------------------------------

SELECT OBJECT Option
:SELECT_OBJECT>
```

Figure A.3



```
********************************************************************************
User  ~gensettle/settle                                                Executive
------------------------------Genesil Version v7.0------------------------------


) START Genesil job ~gensettle/settle/tutor_chip on micro1
) Mon Aug 22 21:08:53 1988
)
)                         Genesil (tm) System Version v7.0
)
)            Copyright   Silicon Compiler Systems Corporation 1988
)
)     Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication   This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation   The reproduction,
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement.
)
)                            RESTRICTED RIGHTS LEGEND
)
)                  Use, duplication or disclosure by the Government
)                     is subject to restrictions as set forth in
)                      subparagraph (c)(1)(ii) of the Rights in
)                    Technical Data and Computer Software clause
)                                at 252 227-7013
)
)
CONTINUE
EXIT_GENESIL
CANCEL
SELECT_OBJECT
ATTACH
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL                         EXISTING
                               NEW

--------------------------------------------------------------------------------

 Enter ATTACH Option
>SELECT_OBJECT>ATTACH>
```

Figure A.4



```
********************************************************************************
User  ~gensettle/settle                                                Executive
-----------------------------Genesil Version v7.0------------------------------


) START Genesil job ~gensettle/settle/tutor_chip on micro1
) Mon Aug 22 21:08:53 1988
)
)                      Genesil (tm) System Version v7.0
)
)           Copyright   Silicon Compiler Systems Corporation 1988
)
)     Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication.  This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation.  The reproduction,
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement.
)
)                         RESTRICTED RIGHTS LEGEND
)
)               Use, duplication or disclosure by the Government
)                 is subject to restrictions as set forth in
)                  subparagraph (c)(1)(ii) of the Rights in
)                Technical Data and Computer Software clause
)                            at 252 227-7013

)
CONTINUE
EXIT_GENESIL
CANCEL
SELECT_OBJECT
ATTACH
NEW
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL           BLOCK           GENERAL_MODULE  CHIP            CHIP_SET
                                 PARALLEL_DP
                                 RANDOM_LOGIC

--------------------------------------------------------------------------------

ATTACH    New Object Type
:SELECT_OBJECT>ATTACH>NEW>
```

Figure A.5

f. Select BACK, which returns the user to the main Executive menu.

g. Now select SELECT_OBJECT, and on the next screen select DOWN. The next screen should indicate a list of sub-objects on the right side of the screen (Figure A.6). Select tutorblk_1 and it will now be attached to the tree.

h. Now go BACK to the initial Executive menu (Figure A.2).

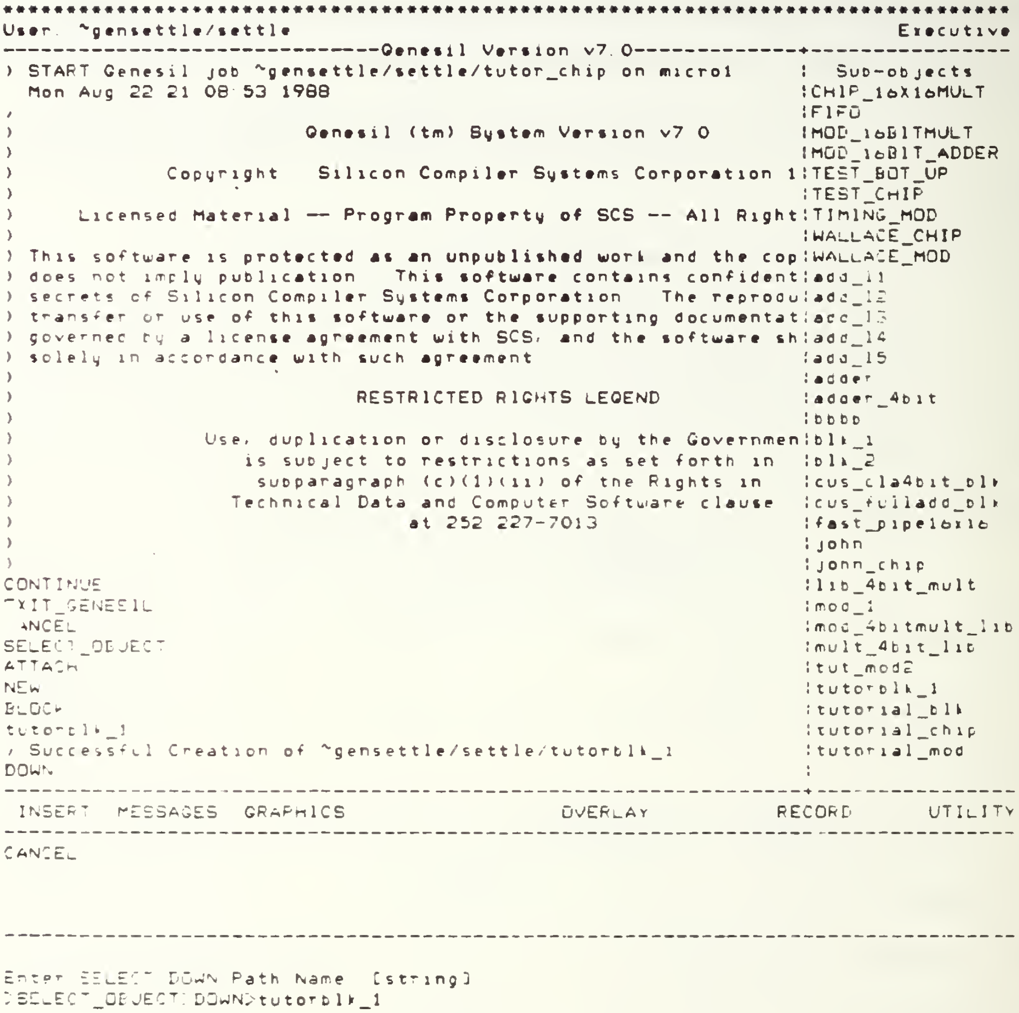
2. The block now needs to be defined:

a. Select DEFINITION (Figure A.2). The next screen is the initial Definition menu (Figure A.7) as denoted by the upper right hand corner of the screen. The upper left hand corner indicates the object types and pathnames.

b. Select HEADER (Figure A.7). The next screen (Figure A.8) is the Header form. Select RANDOM_LOGIC under Function type. CONFIRM it, then select VTC_CP10B under Fab line.

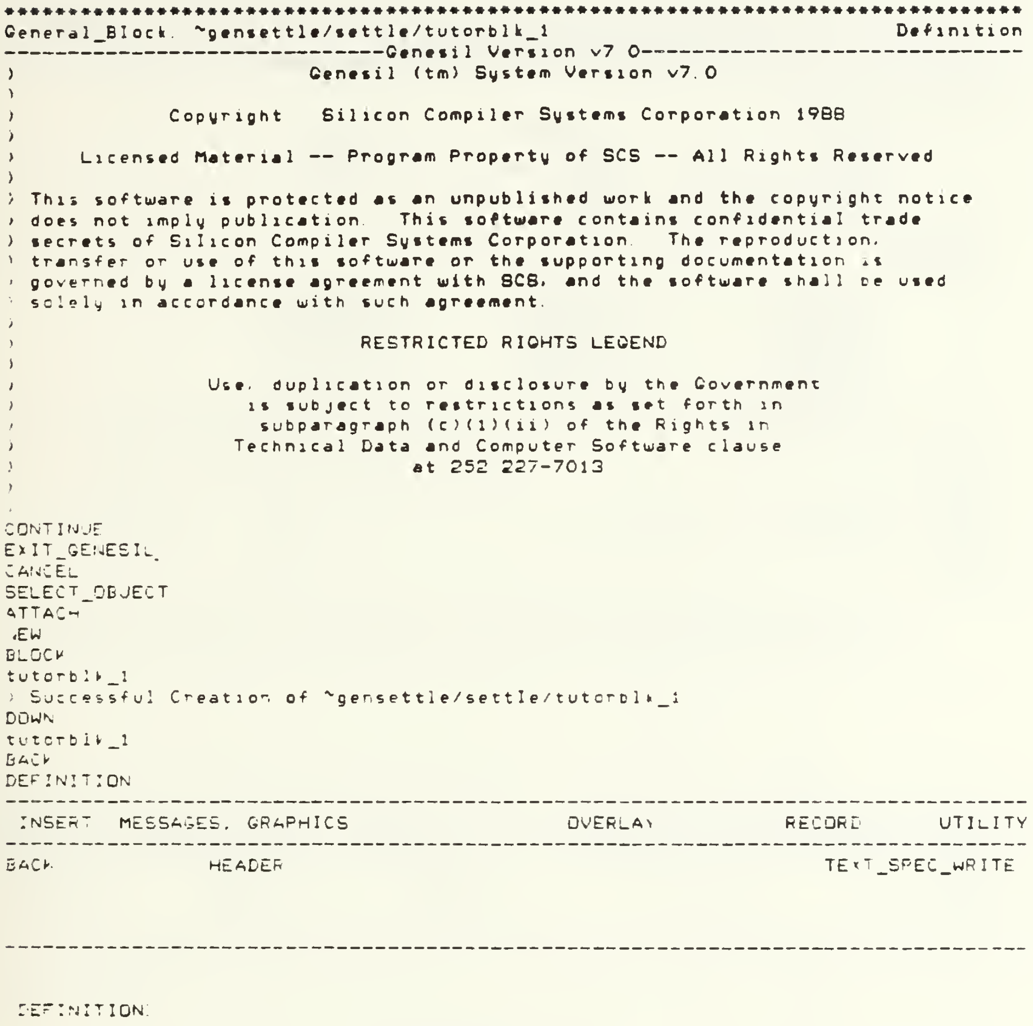
c. Next ACCEPT_FORM (Figure A.8) which will return the screen to the Definition menu. Now select SPECIFICATION which moves the screen to the RANDOM LOGIC Functional Specification form (Figure A.9).

d. Select NEW (Figure A.9), and a random logic library pops up on the right side of the screen. Select ADDER and DFF from the logic library.



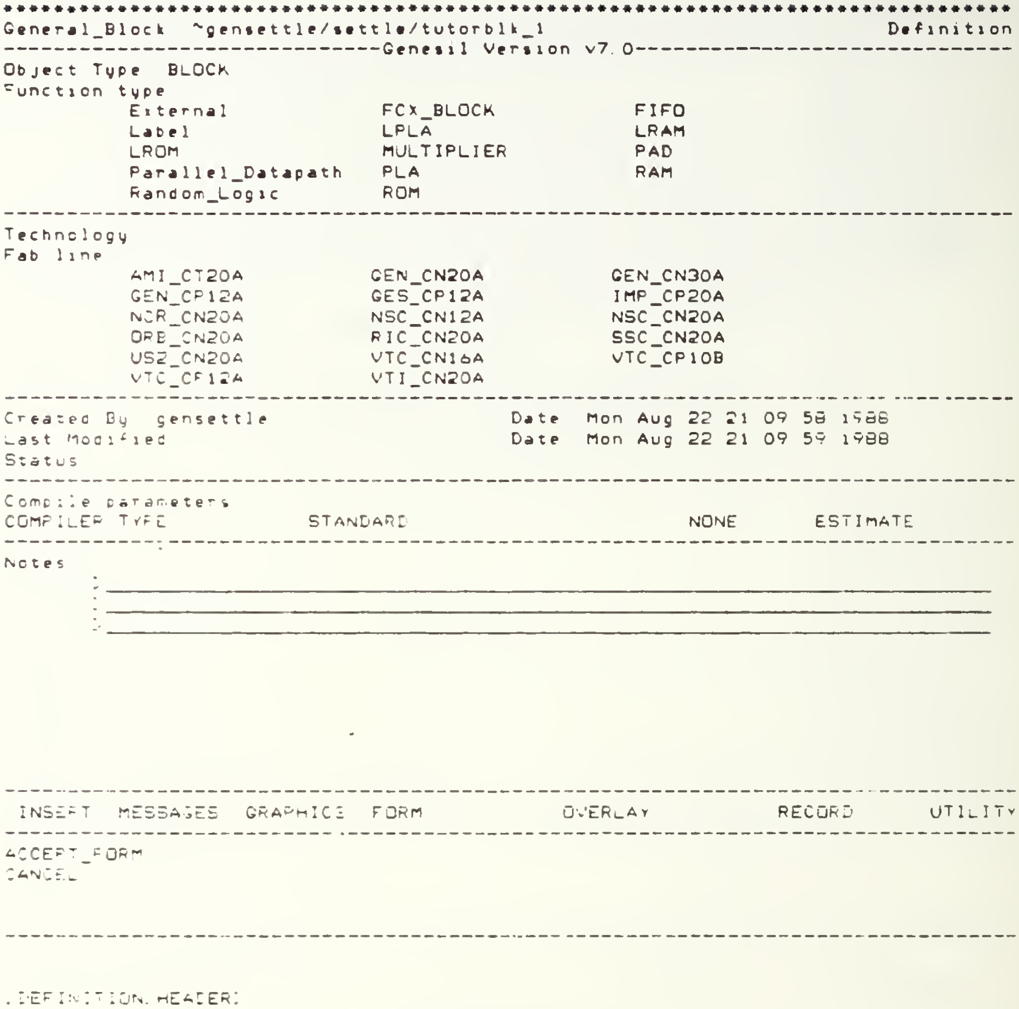
```
********************************************************************************
User. ~gensettle/settle                                                Executive
------------------------------Genesil Version v7.0-------------+----------------
) START Genesil job ~gensettle/settle/tutor_chip on micro1     :  Sub-objects
  Mon Aug 22 21:08:53 1988                                     :CHIP_16X16MULT
,                                                              :FIFO
)                       Genesil (tm) System Version v7.0       :MOD_16BITMULT
)                                                              :MOD_16BIT_ADDER
)           Copyright   Silicon Compiler Systems Corporation 1:TEST_BOT_UP
)                                                              :TEST_CHIP
)     Licensed Material -- Program Property of SCS -- All Right:TIMING_MOD
)                                                              :WALLACE_CHIP
) This software is protected as an unpublished work and the cop:WALLACE_MOD
) does not imply publication. This software contains confident:add_11
) secrets of Silicon Compiler Systems Corporation. The reprodu:add_12
) transfer or use of this software or the supporting documentat:add_13
) governed by a license agreement with SCS, and the software sh:add_14
) solely in accordance with such agreement                     :add_15
)                                                              :adder
)                           RESTRICTED RIGHTS LEGEND           :adder_4bit
)                                                              :bbbb
)              Use, duplication or disclosure by the Governmen:blk_1
)                 is subject to restrictions as set forth in  :blk_2
)                  subparagraph (c)(1)(ii) of the Rights in   :cus_cla4bit_blk
)                Technical Data and Computer Software clause  :cus_fulladd_blk
)                              at 252.227-7013                 :fast_pipe16x16
)                                                              :john
)                                                              :john_chip
CONTINUE                                                       :lib_4bit_mult
EXIT_GENESIL                                                   :mod_1
CANCEL                                                         :mod_4bitmult_lib
SELECT_OBJECT                                                  :mult_4bit_lib
ATTACH                                                         :tut_mod2
NEW                                                            :tutorblk_1
BLOCK                                                          :tutorial_blk
tutorblk_1                                                     :tutorial_chip
/ Successful Creation of ~gensettle/settle/tutorblk_1          :tutorial_mod
DOWN                                                           :
---------------------------------------------------------------+----------------
 INSERT  MESSAGES  GRAPHICS                OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL



--------------------------------------------------------------------------------

Enter SELECT DOWN Path Name  [string]
>SELECT_OBJECT:DOWN>tutorblk_1
```

Figure A.6



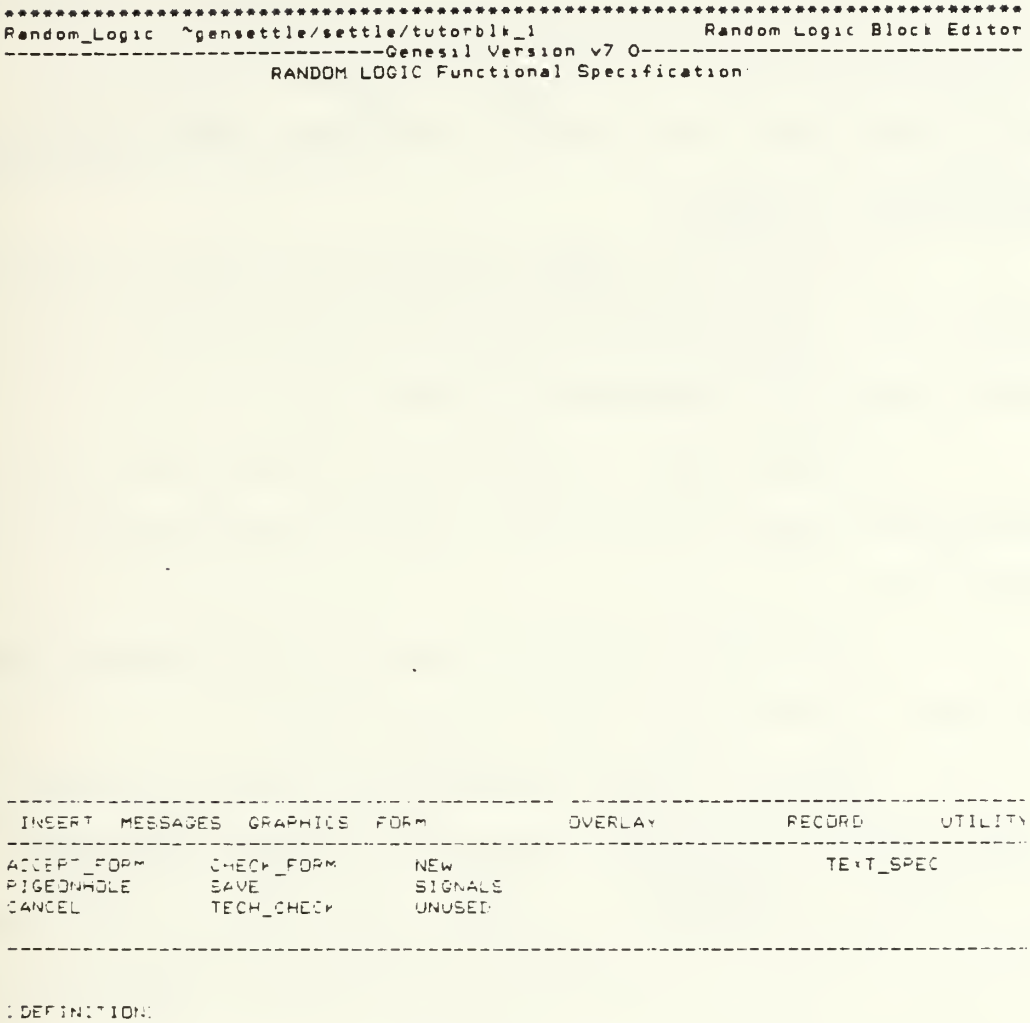
```
*******************************************************************************
General_Block. ~gensettle/settle/tutorblk_1                          Definition
-----------------------------Genesil Version v7.0------------------------------
)                         Genesil (tm) System Version v7.0
)
)            Copyright   Silicon Compiler Systems Corporation 1988
)
)      Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication.  This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation.  The reproduction,
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement.
)
)                            RESTRICTED RIGHTS LEGEND
)
)                  Use, duplication or disclosure by the Government
)                     is subject to restrictions as set forth in
)                      subparagraph (c)(1)(ii) of the Rights in
)                    Technical Data and Computer Software clause
)                                 at 252 227-7013
)
CONTINUE
EXIT_GENESIL
CANCEL
SELECT_OBJECT
ATTACH
NEW
BLOCK
tutorblk_1
) Successful Creation of ~gensettle/settle/tutorblk_1
DOWN
tutorblk_1
BACK
DEFINITION
-------------------------------------------------------------------------------
 INSERT  MESSAGES, GRAPHICS                 OVERLAY          RECORD     UTILITY
-------------------------------------------------------------------------------
BACK            HEADER                                          TEXT_SPEC_WRITE


-------------------------------------------------------------------------------

 DEFINITION:
```

Figure A.7



```
********************************************************************************
General_Block  ~gensettle/settle/tutorblk_1                          Definition
------------------------------Genesil Version v7.0------------------------------
Object Type  BLOCK
Function type
          External            FCX_BLOCK           FIFO
          Label               LPLA                LRAM
          LROM                MULTIPLIER          PAD
          Parallel_Datapath   PLA                 RAM
          Random_Logic        ROM
--------------------------------------------------------------------------------
Technology
Fab line
          AMI_CT20A          GEN_CN20A          GEN_CN30A
          GEN_CP12A          GES_CP12A          IMP_CP20A
          NCR_CN20A          NSC_CN12A          NSC_CN20A
          ORB_CN20A          RIC_CN20A          SSC_CN20A
          USZ_CN20A          VTC_CN16A          VTC_CP10B
          VTC_CP12A          VTI_CN20A
--------------------------------------------------------------------------------
Created By  gensettle                     Date  Mon Aug 22 21 09 58 1988
Last Modified                             Date  Mon Aug 22 21 09 59 1988
Status
--------------------------------------------------------------------------------
Compile parameters
COMPILER TYPE           STANDARD                      NONE      ESTIMATE
--------------------------------------------------------------------------------
Notes
        :______________________________________________________________________
        :______________________________________________________________________
        :______________________________________________________________________



--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM
CANCEL

--------------------------------------------------------------------------------

[DEFINITION.HEADER]
```

Figure A.8



```
*******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1               Random Logic Block Editor
-----------------------------Genesil Version v7 0--------------------------------
                    RANDOM LOGIC Functional Specification



------------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------------
ACCEPT_FORM       CHECK_FORM       NEW                           TEXT_SPEC
PIGEONHOLE        SAVE             SIGNALS
CANCEL            TECH_CHECK       UNUSED

------------------------------------------------------------------------------------

:DEFINITION:
```

Figure A.9

e. Now select BACK and the screen should be back in the RANDOM LOGIC Functional Specification form (Figure A.10) with the ADDER and DFF included.

f. Select EDIT, which is adjacent to ADDER0 (Figure A.10) and the next screen will be a specification form for the adder (Figure A.11). Details of all random logic specification forms are found in the Genesil Silicon Compiler Library Vol I, Blocks.

g. Fill in the adder specification form as shown in Figure A.12. Select EXPAND for a line-by-line entry form if desired and select COMPRESS to return.

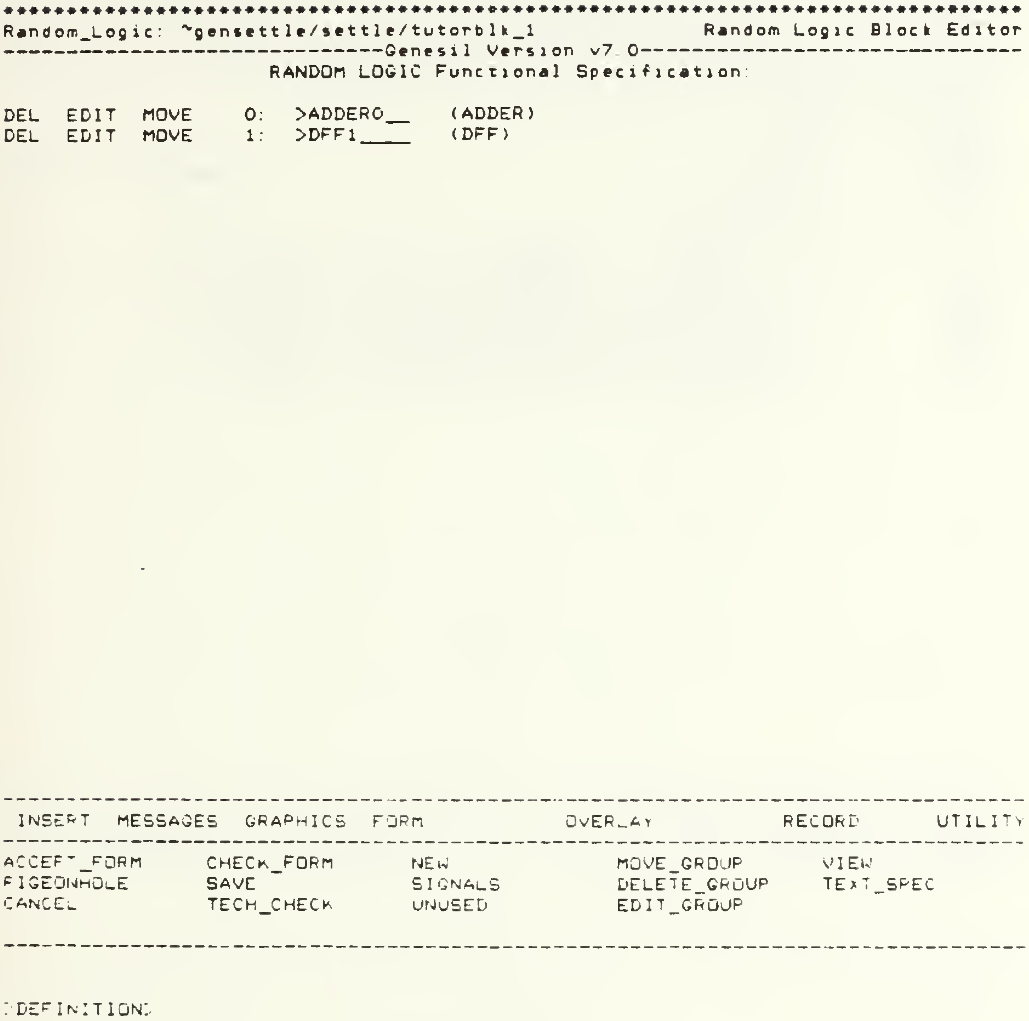
h. Select NEXT (Figure A.12), which pulls up a specification menu for the DFF. Fill it out as shown in Figure A.13.

i. Now select BACK to return to the RANDOM LOGIC Functional Specification form (Figure A.10).

j. Select SIGNALS (Figure A.10) and the screen shows a signal list of the block. Make the signals correspond to Figure A.14 by selecting I, O and L next to the signal names. This cleans up the circuit because the system assumes the user desires Both normally.

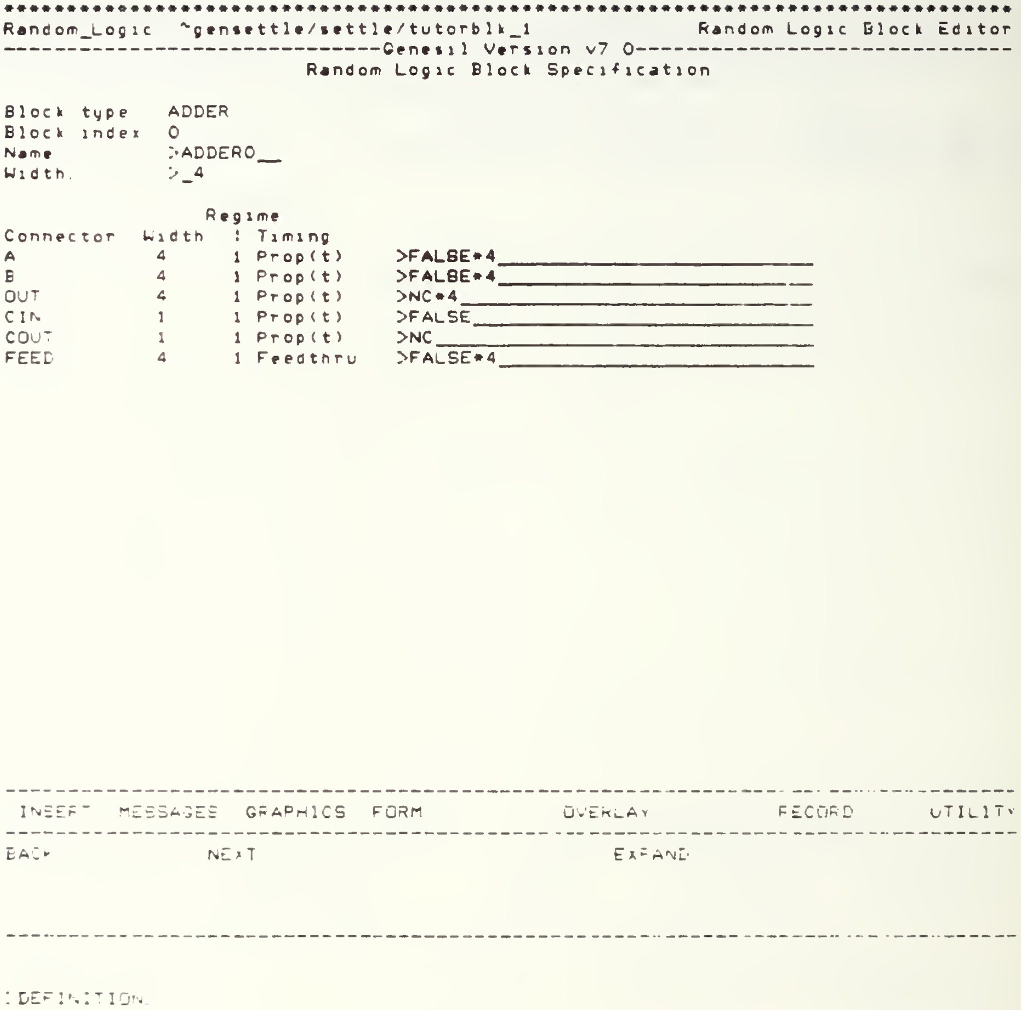
k. Now select BACK to return to Figure A.10.

l. If desired, VIEW may now be selected for a block diagram, with signals, for inspection. Use BACK to return to the specification form.



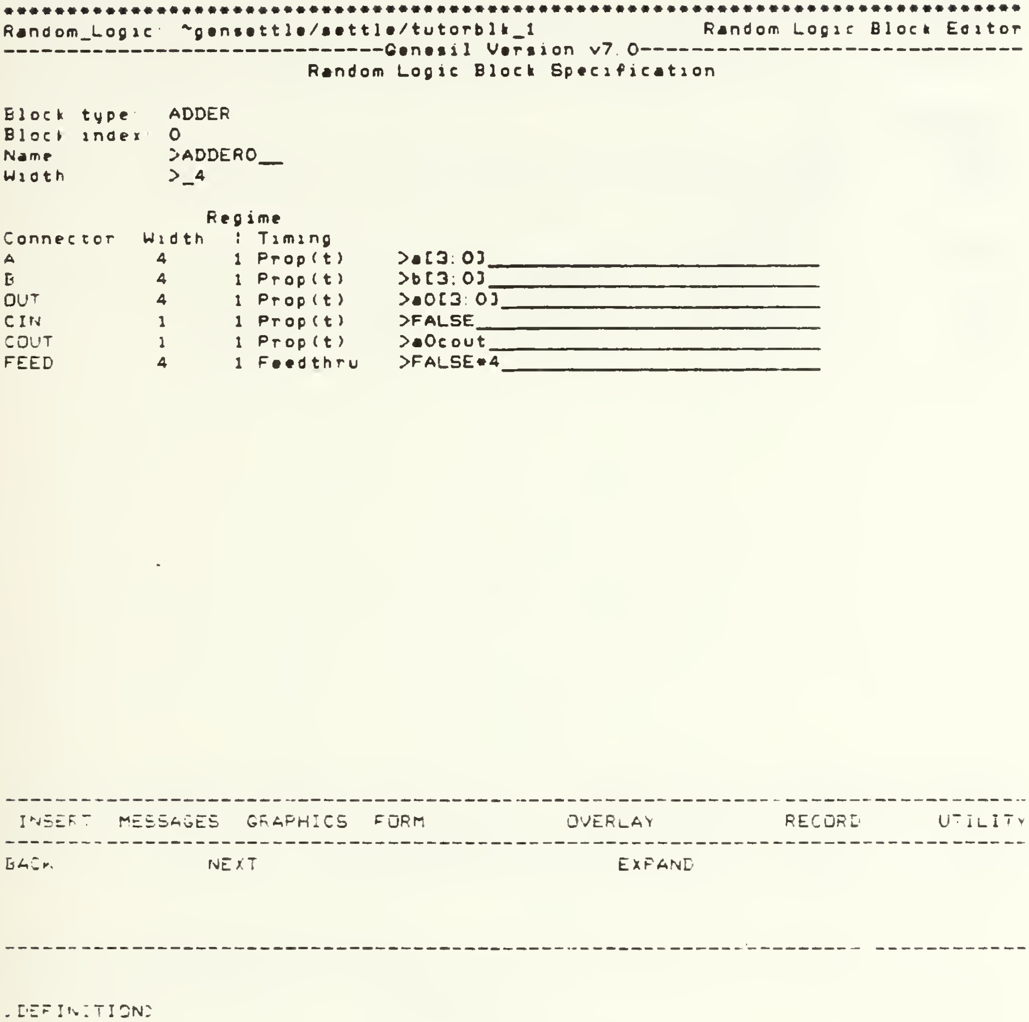
```
*****************************************************************************
Random_Logic: ~gensettle/settle/tutorblk_1              Random Logic Block Editor
------------------------------Genesil Version v7.0------------------------------
                    RANDOM LOGIC Functional Specification:

DEL  EDIT  MOVE    0:  >ADDER0__    (ADDER)
DEL  EDIT  MOVE    1:  >DFF1____    (DFF)
















--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_FORM      NEW            MOVE_GROUP       VIEW
PIGEONHOLE       SAVE            SIGNALS        DELETE_GROUP     TEXT_SPEC
CANCEL           TECH_CHECK      UNUSED         EDIT_GROUP

--------------------------------------------------------------------------------

<DEFINITION>
```

Figure A.10



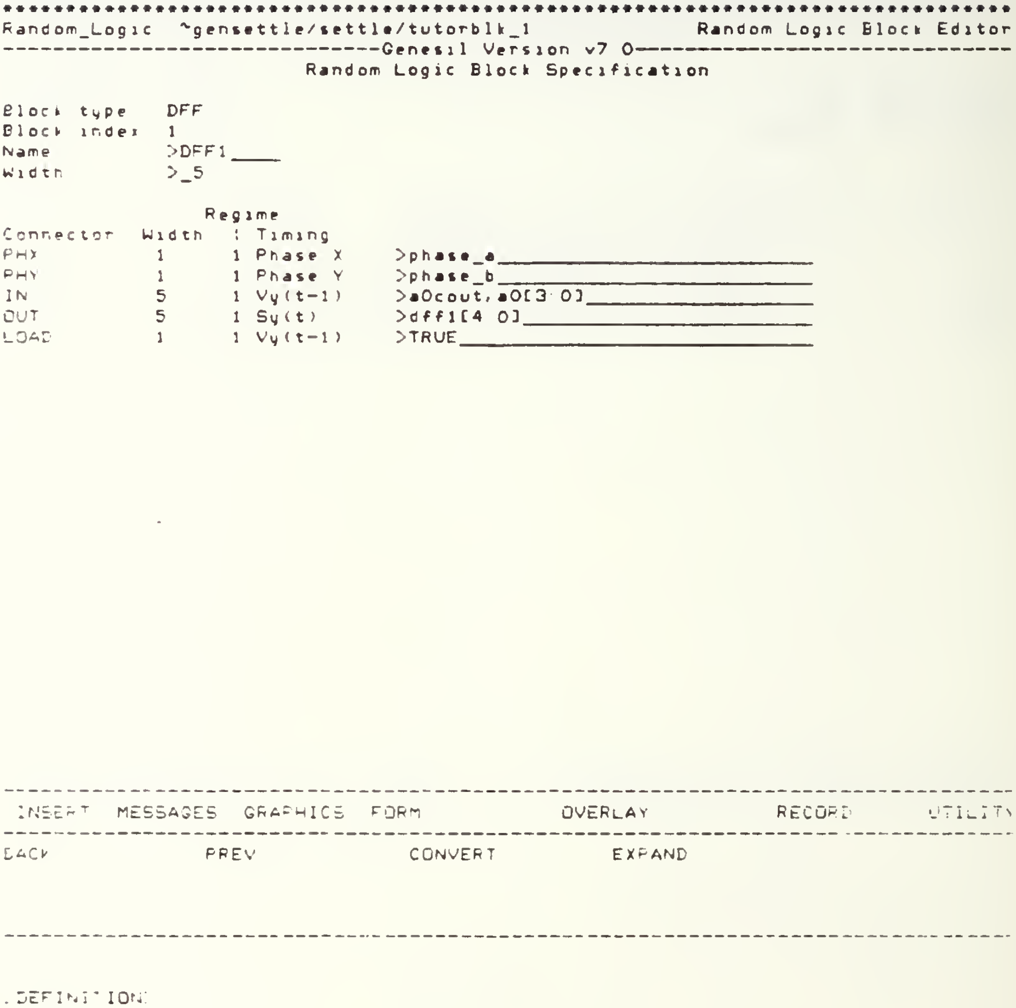
```
*************************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1              Random Logic Block Editor
-------------------------------Genesil Version v7 0-------------------------------
                                Random Logic Block Specification

Block type    ADDER
Block index   0
Name          >ADDER0__
Width.        >_4

                    Regime
Connector  Width  : Timing
A            4      1 Prop(t)      >FALBE*4________________________
B            4      1 Prop(t)      >FALBE*4________________________
OUT          4      1 Prop(t)      >NC*4___________________________
CIN          1      1 Prop(t)      >FALSE__________________________
COUT         1      1 Prop(t)      >NC_____________________________
FEED         4      1 Feedthru     >FALSE*4________________________
```

Figure A.11



```
********************************************************************************
Random_Logic: ~gensettle/settle/tutorblk_1              Random Logic Block Editor
------------------------------Genesil Version v7.0------------------------------
                         Random Logic Block Specification

Block type:    ADDER
Block index:   0
Name           >ADDER0__
Width          >_4

                 Regime
Connector  Width   : Timing
A            4     1 Prop(t)     >a[3:0]_________________________
B            4     1 Prop(t)     >b[3:0]_________________________
OUT          4     1 Prop(t)     >a0[3:0]________________________
CIN          1     1 Prop(t)     >FALSE__________________________
COUT         1     1 Prop(t)     >a0cout_________________________
FEED         4     1 Feedthru    >FALSE*4________________________





--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK            NEXT                             EXPAND


--------------------------------------------------------------------------------

<DEFINITION>
```

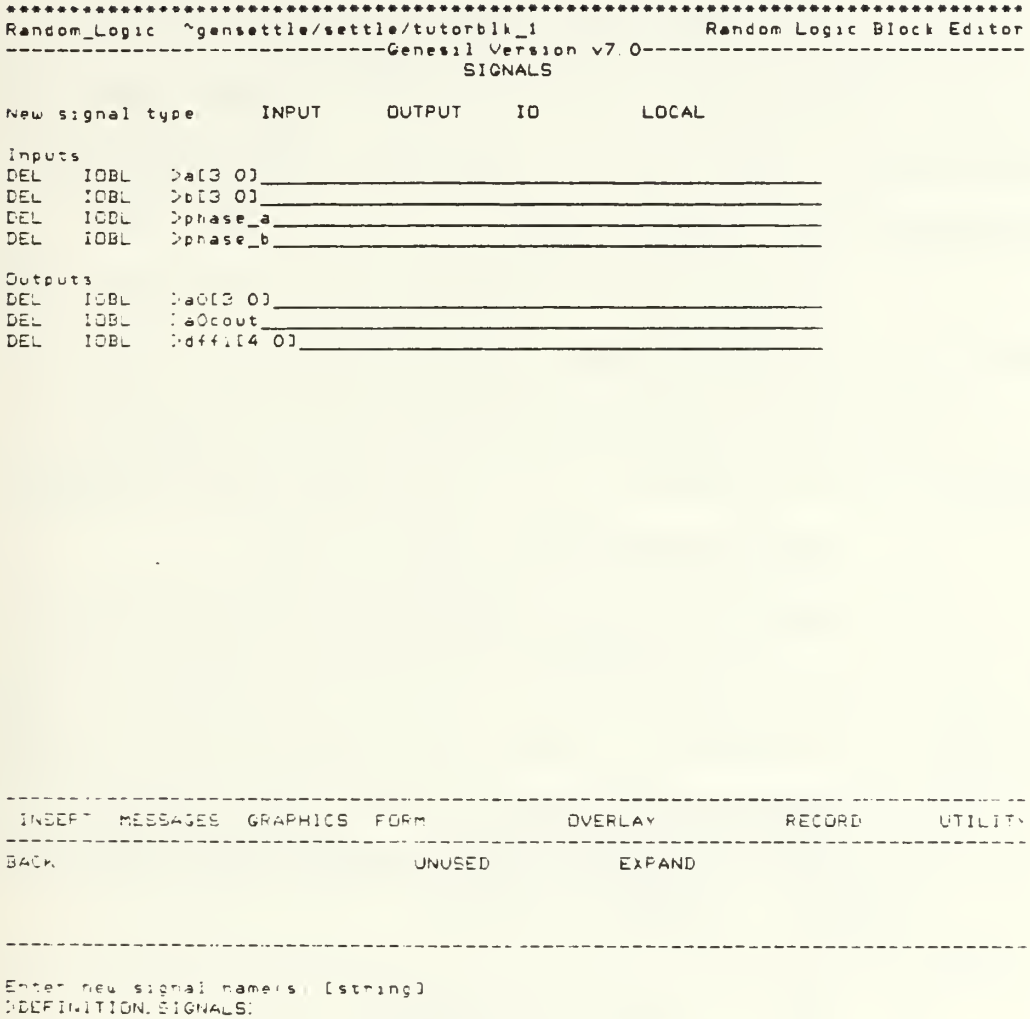
Figure A.12



```
******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1              Random Logic Block Editor
-----------------------------Genesil Version v7 0---------------------------------
                          Random Logic Block Specification

Block type    DFF
Block index   1
Name          >DFF1____
Width         >_5

                Regime
Connector  Width   : Timing
PHX          1     1 Phase X      >phase_a________________________________
PHY          1     1 Phase Y      >phase_b________________________________
IN           5     1 Vy(t-1)      >a0cout,a0[3 0]_________________________
OUT          5     1 Sy(t)        >dff1[4 0]______________________________
LOAD         1     1 Vy(t-1)      >TRUE___________________________________
```

```
----------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD        UTILITY
----------------------------------------------------------------------------------
BACK            PREV            CONVERT            EXPAND


----------------------------------------------------------------------------------

.DEFINITION
```

Figure A.13



```
*************************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                    Random Logic Block Editor
------------------------------Genesil Version v7.0------------------------------
                                    SIGNALS

New signal type      INPUT      OUTPUT     IO         LOCAL

Inputs
DEL    IOBL    >a[3 0]__________________________________________
DEL    IOBL    >b[3 0]__________________________________________
DEL    IOBL    >phase_a_________________________________________
DEL    IOBL    >phase_b_________________________________________

Outputs
DEL    IOBL    >a0[3 0]_________________________________________
DEL    IOBL    >a0cout__________________________________________
DEL    IOBL    >dffi[4 0]_______________________________________









--------------------------------------------------------------------------------
 INSERT   MESSAGES   GRAPHICS   FORM           OVERLAY          RECORD      UTILIT
--------------------------------------------------------------------------------
BACK                                UNUSED           EXPAND


--------------------------------------------------------------------------------

Enter new signal name(s) [string]
>DEFINITION.SIGNALS:
```

Figure A.14

m. Now select ACCEPT_FORM, and after the system writes the text file and validates the form, the screen should be back to the Executive menu (Figure A.15).

3. The block will now be compiled:

a. From Figure A.15 select COMPILE. The next screen gives the user various compile options including simulation, timing analysis, and layout. Since all will be used later, select BUILD_ALL (Figure A.16).

b. At completion of compile, the screen will again return to the Executive menu (Figure A.17).

4. The block will now be simulated. Detailed simulation information can be found in the Simulation Users Guide:

a. Select SIMULATION (Figure A.17).

b. Select GFL then SIMULATE on the Simulation Environment form (Figure A.18). Notice the system is now in the Functional Simulator (upper right hand corner of screen).

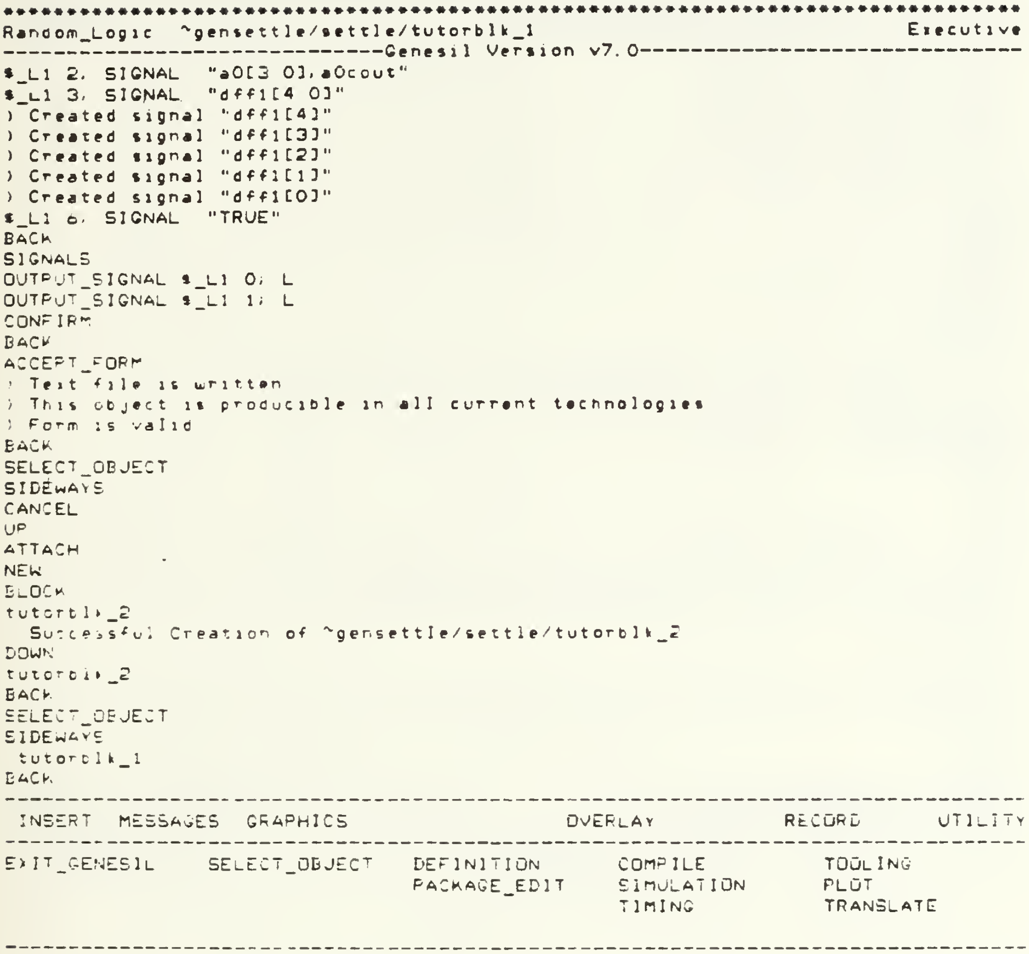
c. Select BIND (Figure A.19) to input signal values.

d. Select MULTIPLE_SIGS (Figure A.20) since there are several signals to input.

e. Type in a[0] etc., and the value (0 or 1) as prompted. Use the values shown in Figure A.20 as initial examples.

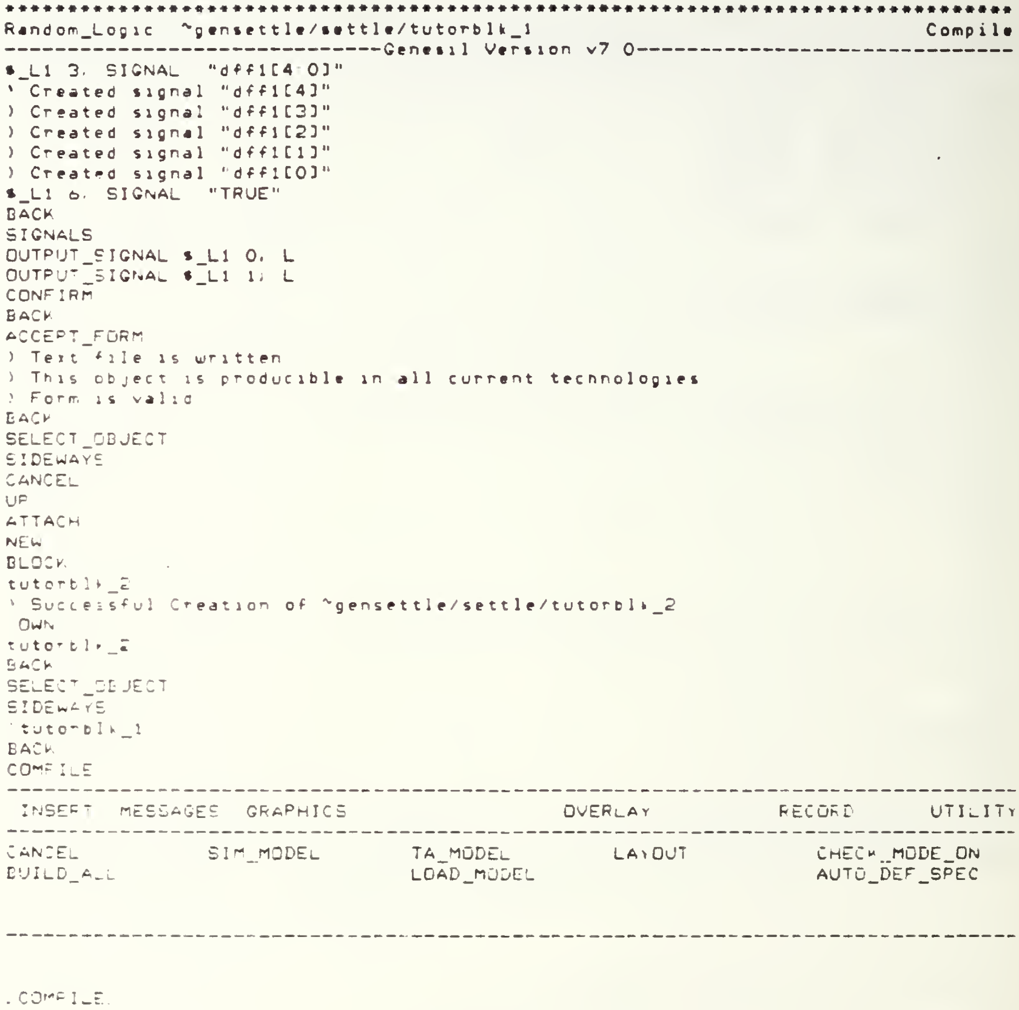
f. When all desired signal values have been entered, select BACK (Figure A.20).

g. Now select CYCLE and 2 (Figure A.21) to cycle the system clocks twice.



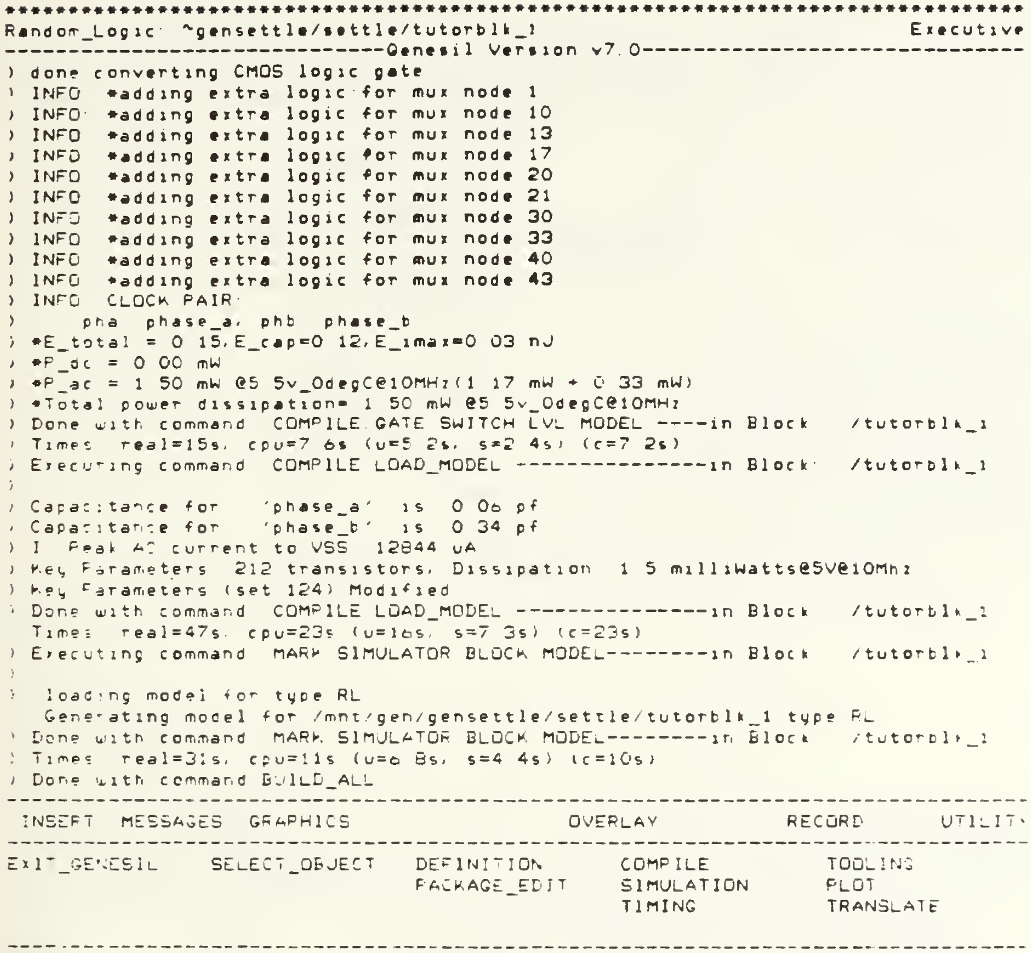
```
********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                              Executive
--------------------------------Genesil Version v7.0------------------------------
$_L1 2, SIGNAL  "a0[3 0],a0cout"
$_L1 3, SIGNAL  "dff1[4 0]"
) Created signal "dff1[4]"
) Created signal "dff1[3]"
) Created signal "dff1[2]"
) Created signal "dff1[1]"
) Created signal "dff1[0]"
$_L1 6, SIGNAL  "TRUE"
BACK
SIGNALS
OUTPUT_SIGNAL $_L1 0; L
OUTPUT_SIGNAL $_L1 1; L
CONFIRM
BACK
ACCEPT_FORM
) Text file is written
) This object is producible in all current technologies
) Form is valid
BACK
SELECT_OBJECT
SIDEWAYS
CANCEL
UP
ATTACH
NEW
BLOCK
tutorblk_2
  Successful Creation of ~gensettle/settle/tutorblk_2
DOWN
tutorblk_2
BACK
SELECT_OBJECT
SIDEWAYS
 tutorblk_1
BACK
---------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                         OVERLAY          RECORD      UTILITY
---------------------------------------------------------------------------------
EXIT_GENESIL     SELECT_OBJECT    DEFINITION       COMPILE          TOOLING
                                  PACKAGE_EDIT     SIMULATION       PLOT
                                                   TIMING           TRANSLATE

---------------------------------------------------------------------------------
```

Figure A,15



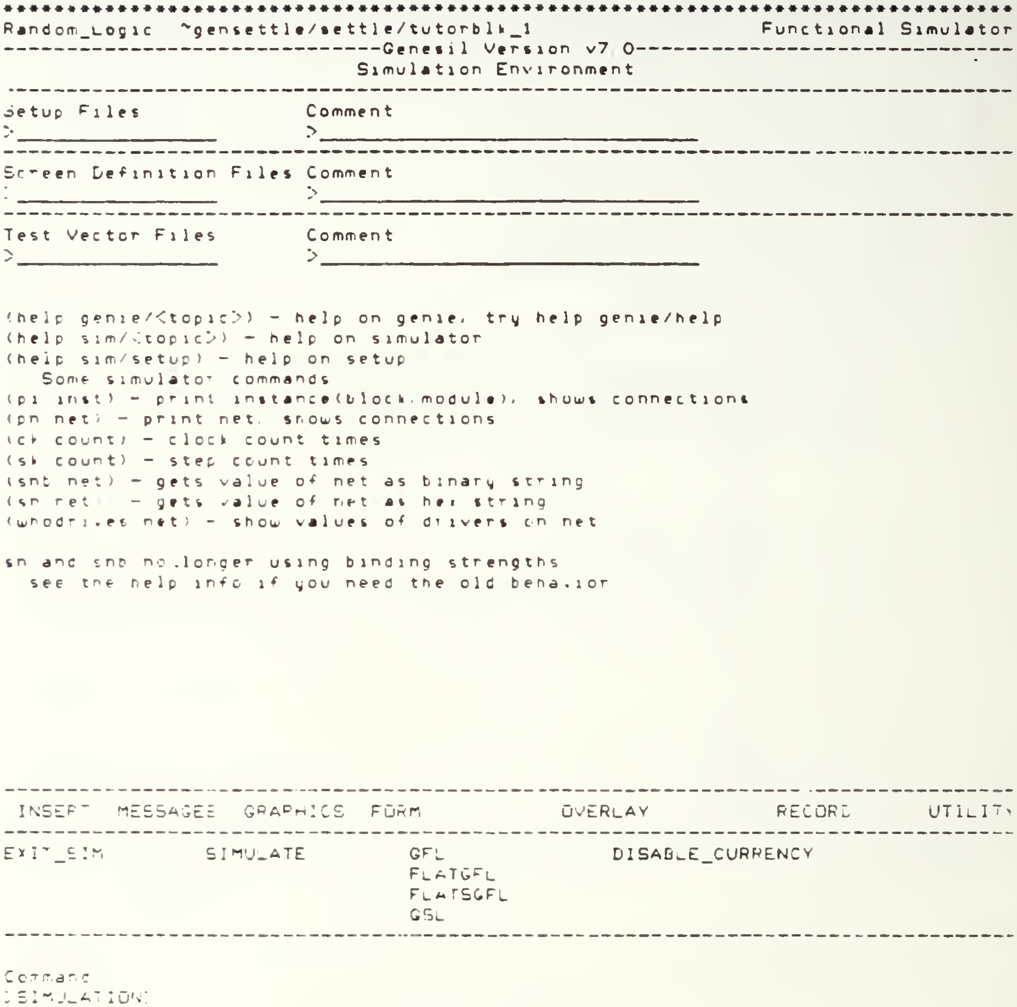
```
************************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                                   Compile
----------------------------Genesil Version v7 0-----------------------------
$_L1 3, SIGNAL  "dff1[4:0]"
) Created signal "dff1[4]"
) Created signal "dff1[3]"
) Created signal "dff1[2]"
) Created signal "dff1[1]"
) Created signal "dff1[0]"
$_L1 6, SIGNAL  "TRUE"
BACK
SIGNALS
OUTPUT_SIGNAL $_L1 0, L
OUTPUT_SIGNAL $_L1 1, L
CONFIRM
BACK
ACCEPT_FORM
) Text file is written
) This object is producible in all current technologies
) Form is valid
BACK
SELECT_OBJECT
SIDEWAYS
CANCEL
UP
ATTACH
NEW
BLOCK
tutorblk_2
) Successful Creation of ~gensettle/settle/tutorblk_2
 OWN
tutorblk_2
BACK
SELECT_OBJECT
SIDEWAYS
 tutorblk_1
BACK
COMPILE
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                      OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL          SIM_MODEL       TA_MODEL        LAYOUT          CHECK_MODE_ON
BUILD_ALL                       LOAD_MODEL                      AUTO_DEF_SPEC


--------------------------------------------------------------------------------

.COMPILE.
```

Figure A.16



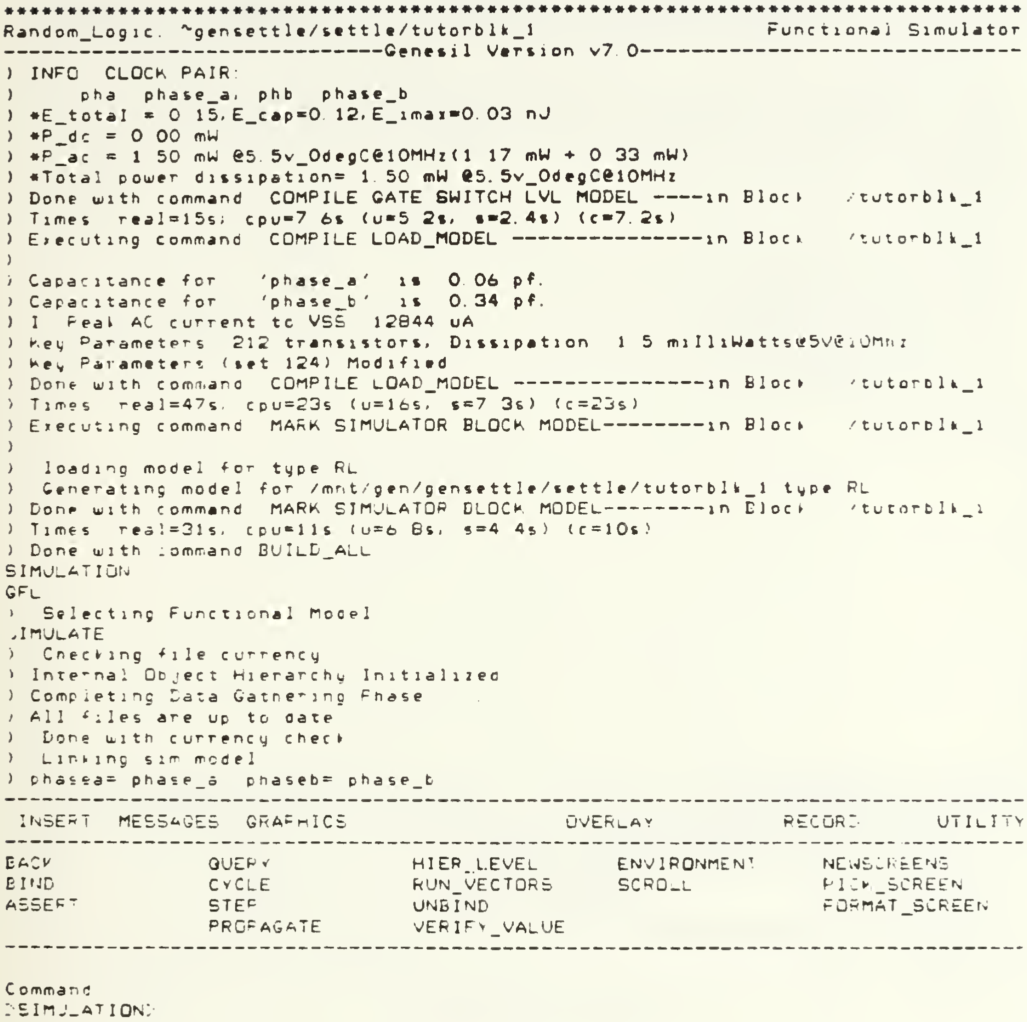
```
*************************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                                 Executive
--------------------------------Genesil Version v7 0--------------------------------
) done converting CMOS logic gate
) INFO  *adding extra logic for mux node 1
) INFO  *adding extra logic for mux node 10
) INFO  *adding extra logic for mux node 13
) INFO  *adding extra logic for mux node 17
) INFO  *adding extra logic for mux node 20
) INFO  *adding extra logic for mux node 21
) INFO  *adding extra logic for mux node 30
) INFO  *adding extra logic for mux node 33
) INFO  *adding extra logic for mux node 40
) INFO  *adding extra logic for mux node 43
) INFO  CLOCK PAIR
)     pha  phase_a, phb  phase_b
) *E_total = 0 15,E_cap=0 12,E_imax=0 03 nJ
) *P_dc = 0 00 mW
) *P_ac = 1 50 mW @5 5v_0degC@10MHz(1 17 mW + 0 33 mW)
) *Total power dissipation= 1 50 mW @5 5v_0degC@10MHz
) Done with command  COMPILE GATE SWITCH LVL MODEL ----in Block   /tutorblk_1
) Times  real=15s, cpu=7 6s (u=5 2s, s=2 4s) (c=7 2s)
) Executing command  COMPILE LOAD_MODEL ----------------in Block   /tutorblk_1
)
) Capacitance for   'phase_a'  is  0 06 pf
) Capacitance for   'phase_b'  is  0 34 pf
) I  Peak AC current to VSS  12844 uA
) Key Parameters  212 transistors, Dissipation  1 5 milliWatts@5V@10Mhz
) Key Parameters (set 124) Modified
) Done with command  COMPILE LOAD_MODEL ----------------in Block   /tutorblk_1
  Times  real=47s, cpu=23s (u=16s, s=7 3s) (c=23s)
) Executing command  MARK SIMULATOR BLOCK MODEL---------in Block   /tutorblk_1
)
)  loading model for type RL
   Generating model for /mnt/gen/gensettle/settle/tutorblk_1 type RL
) Done with command  MARK SIMULATOR BLOCK MODEL---------in Block   /tutorblk_1
) Times  real=31s, cpu=11s (u=6 8s, s=4 4s) (c=10s)
) Done with command BUILD_ALL
-------------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                        OVERLAY          RECORD      UTILITY
-------------------------------------------------------------------------------------
EXIT_GENESIL     SELECT_OBJECT     DEFINITION       COMPILE          TOOLING
                                   PACKAGE_EDIT     SIMULATION       PLOT
                                                    TIMING           TRANSLATE

-------------------------------------------------------------------------------------
```

Figure A.17



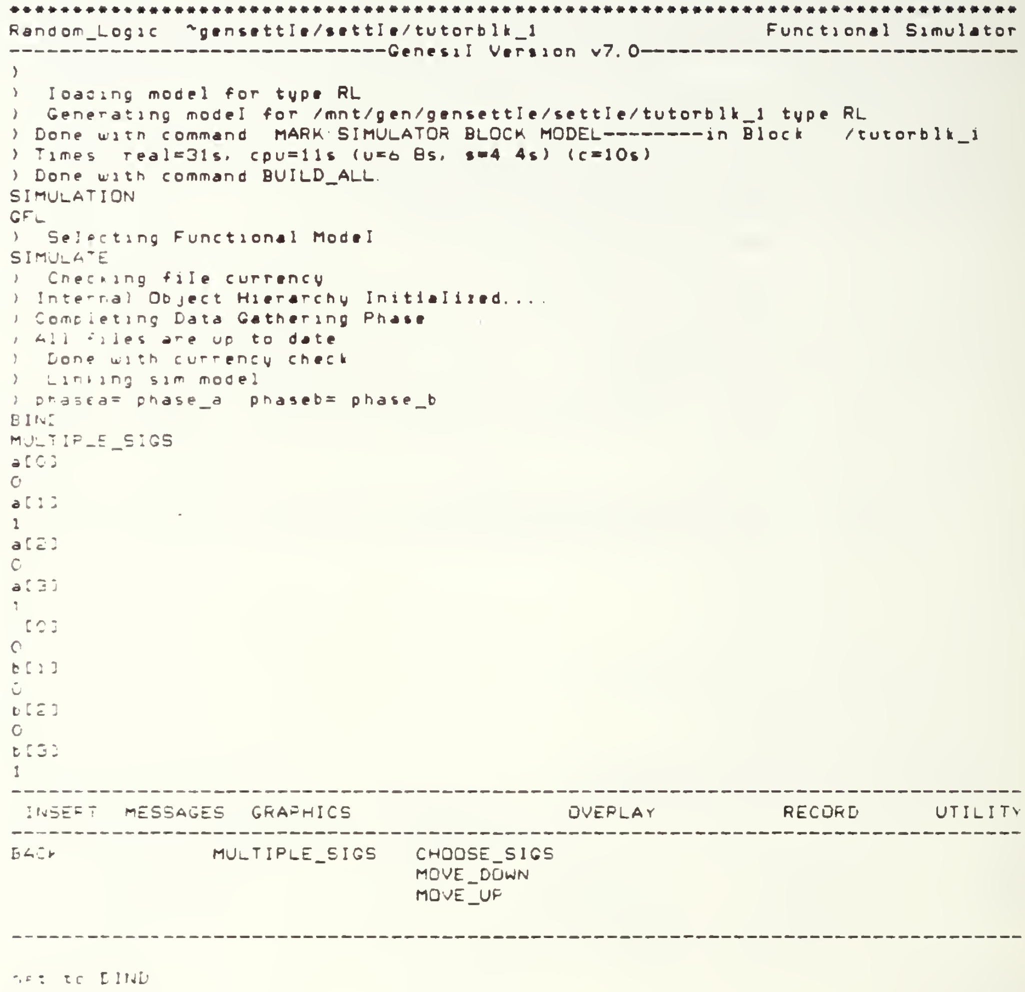
```
********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                   Functional Simulator
-------------------------------Genesil Version v7.0------------------------------
                                Simulation Environment
---------------------------------------------------------------------------------
Setup Files              Comment
>________________        >_________________________________
---------------------------------------------------------------------------------
Screen Definition Files  Comment
:________________        >_________________________________
---------------------------------------------------------------------------------
Test Vector Files        Comment
>________________        >_________________________________


(help genie/<topic>) - help on genie, try help genie/help
(help sim/<topic>) - help on simulator
(help sim/setup) - help on setup
   Some simulator commands
(pi inst) - print instance(block,module), shows connections
(pn net) - print net, shows connections
(ck count) - clock count times
(sk count) - step count times
(snb net) - gets value of net as binary string
(sn net)  - gets value of net as hex string
(whodrives net) - show values of drivers on net

sn and snb no longer using binding strengths
  see the help info if you need the old behavior





---------------------------------------------------------------------------------
 INSERT   MESSAGES   GRAPHICS   FORM           OVERLAY          RECORD      UTILITY
---------------------------------------------------------------------------------
EXIT_SIM         SIMULATE         GFL               DISABLE_CURRENCY
                                  FLATGFL
                                  FLATSGFL
                                  GSL
---------------------------------------------------------------------------------

Command
[SIMULATION]
```

Figure A.18



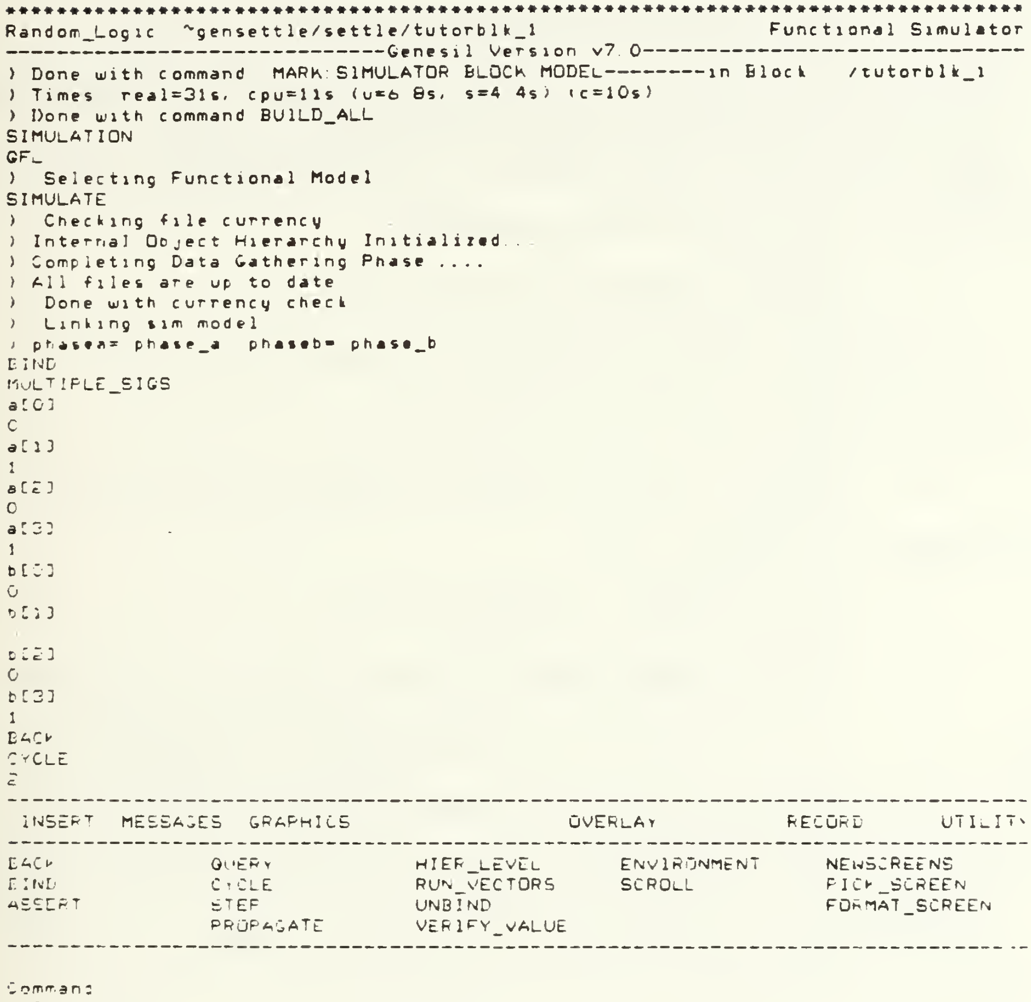
```
********************************************************************************
Random_Logic. ~gensettle/settle/tutorblk_1                   Functional Simulator
------------------------------Genesil Version v7.0-------------------------------
) INFO  CLOCK PAIR:
)      pha  phase_a, phb  phase_b
) *E_total = 0 15,E_cap=0 12,E_imax=0.03 nJ
) *P_dc = 0 00 mW
) *P_ac = 1 50 mW @5.5v_OdegC@10MHz(1 17 mW + 0 33 mW)
) *Total power dissipation= 1.50 mW @5.5v_OdegC@10MHz
) Done with command  COMPILE GATE SWITCH LVL MODEL ----in Block   /tutorblk_1
) Times  real=15s; cpu=7 6s (u=5 2s, s=2.4s) (c=7.2s)
) Executing command  COMPILE LOAD_MODEL ----------------in Block   /tutorblk_1
)
; Capacitance for   'phase_a'  is  0.06 pf.
) Capacitance for   'phase_b'  is  0.34 pf.
) I  Peak AC current to VSS  12844 uA
) Key Parameters  212 transistors, Dissipation  1 5 milliWatts@5V@10MHz
) Key Parameters (set 124) Modified
) Done with command  COMPILE LOAD_MODEL ----------------in Block   /tutorblk_1
) Times  real=47s. cpu=23s (u=16s, s=7 3s) (c=23s)
) Executing command  MARK SIMULATOR BLOCK MODEL--------in Block   /tutorblk_1
)
)  loading model for type RL
)  Generating model for /mnt/gen/gensettle/settle/tutorblk_1 type RL
) Done with command  MARK SIMULATOR BLOCK MODEL--------in Block   /tutorblk_1
) Times  real=31s, cpu=11s (u=6 8s, s=4 4s) (c=10s)
) Done with command BUILD_ALL
SIMULATION
GFL
)  Selecting Functional Model
SIMULATE
)  Checking file currency
) Internal Object Hierarchy Initialized
) Completing Data Gathering Phase
) All files are up to date
)  Done with currency check
)  Linking sim model
) phasea= phase_a  phaseb= phase_b
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
BACK            QUERY           HIER_LEVEL      ENVIRONMENT     NEWSCREENS
BIND            CYCLE           RUN_VECTORS     SCROLL          PICK_SCREEN
ASSERT          STEP            UNBIND                          FORMAT_SCREEN
                PROPAGATE       VERIFY_VALUE
--------------------------------------------------------------------------------

Command
>SIMULATION>
```

Figure A.19



```
*********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                    Functional Simulator
------------------------------Genesil Version v7.0-------------------------------
)
)  Loading model for type RL
)  Generating model for /mnt/gen/gensettle/settle/tutorblk_1 type RL
) Done with command  MARK SIMULATOR BLOCK MODEL---------in Block   /tutorblk_1
) Times  real=31s, cpu=11s (u=6 8s, s=4 4s) (c=10s)
) Done with command BUILD_ALL.
SIMULATION
GFL
)  Selecting Functional Model
SIMULATE
)  Checking file currency
) Internal Object Hierarchy Initialized....
) Completing Data Gathering Phase
) All files are up to date
)  Done with currency check
)  Linking sim model
) phasea= phase_a  phaseb= phase_b
. BIND
MULTIPLE_SIGS
a[0]
0
a[1]
1
a[2]
0
a[3]
1
 [0]
0
b[1]
0
b[2]
0
b[3]
1
---------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                    OVERLAY           RECORD      UTILITY
---------------------------------------------------------------------------------
BACK            MULTIPLE_SIGS   CHOOSE_SIGS
                                MOVE_DOWN
                                MOVE_UP

---------------------------------------------------------------------------------

Set to BIND
 SIMULATION
```

Figure A.20



```
*******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                  Functional Simulator
-----------------------------Genesil Version v7.0-------------------------------
) Done with command  MARK:SIMULATOR BLOCK MODEL---------in Block   /tutorblk_1
) Times  real=31s, cpu=11s (u=6.8s, s=4.4s) (c=10s)
) Done with command BUILD_ALL
SIMULATION
GFL
)  Selecting Functional Model
SIMULATE
)  Checking file currency
) Internal Object Hierarchy Initialized...
) Completing Data Gathering Phase ....
) All files are up to date
)  Done with currency check
)  Linking sim model
) phasea= phase_a  phaseb= phase_b
BIND
MULTIPLE_SIGS
a[0]
0
a[1]
1
a[2]
0
a[3]
1
b[0]
0
b[1]
1
b[2]
0
b[3]
1
BACK
CYCLE
2
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                    OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK            QUERY           HIER_LEVEL      ENVIRONMENT     NEWSCREENS
BIND            CYCLE           RUN_VECTORS     SCROLL          PICK_SCREEN
ASSERT          STEP            UNBIND                          FORMAT_SCREEN
                PROPAGATE       VERIFY_VALUE
--------------------------------------------------------------------------------

Command:
.SIMULATION: p1
```

Figure A.21

h. Type in pi and depress RETURN (Figure A.21).

i. Now the screen should look like Figure A.22, which has simulated the block for proper logic. Other values may be inserted by using the previous steps beginning with c.

j. Now select BACK, then EXIT_SIM, with a CONFIRM to return to the Executive menu (Figure A.17)

5. Timing analysis will now be performed on the block. The Timing Analysis Users Guide contains detailed information concerning timing data and commands:

a. Select TIMING (Figure A.17).

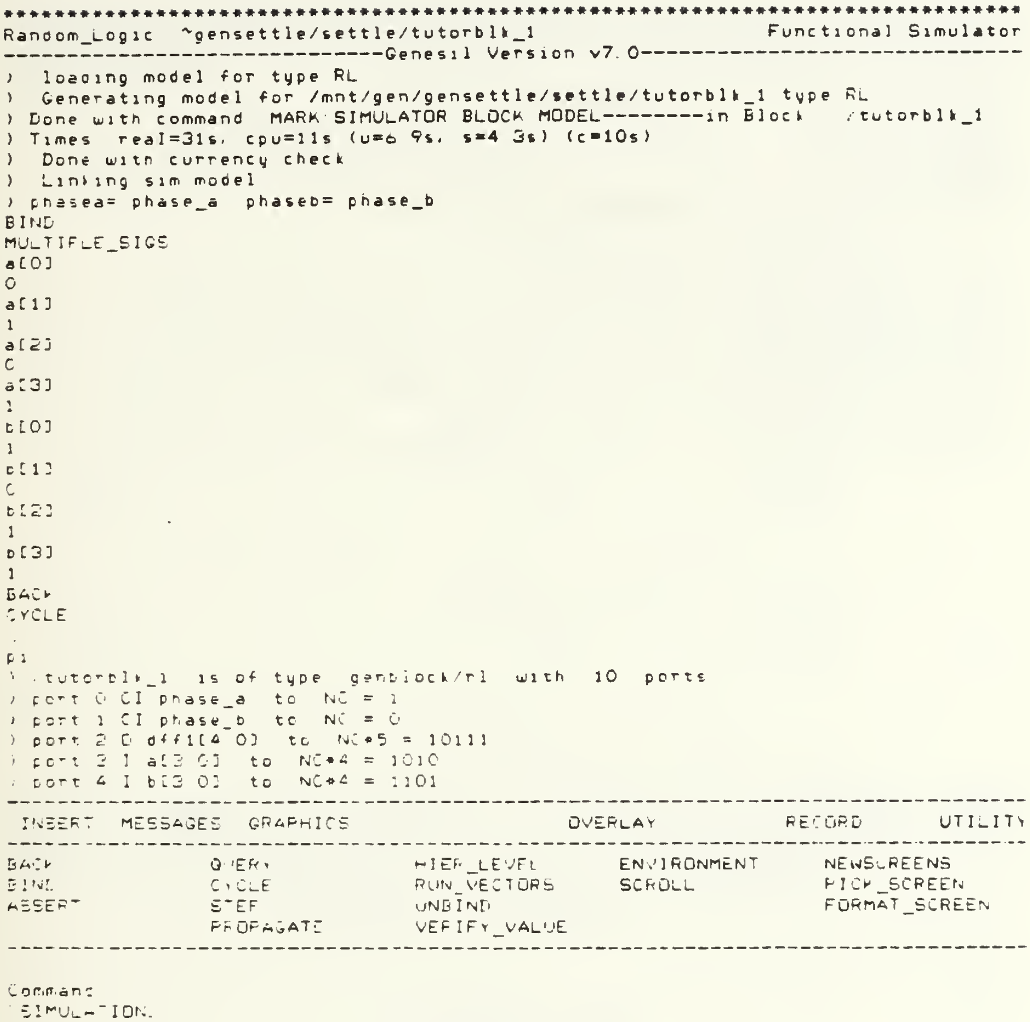
b. The system should be in the Timing Analyser function as shown in Figure A.23.

c. Select CLOCKS, and all object clock information is as shown in Figure A.24.

d. Select BACK (Figure A.24).

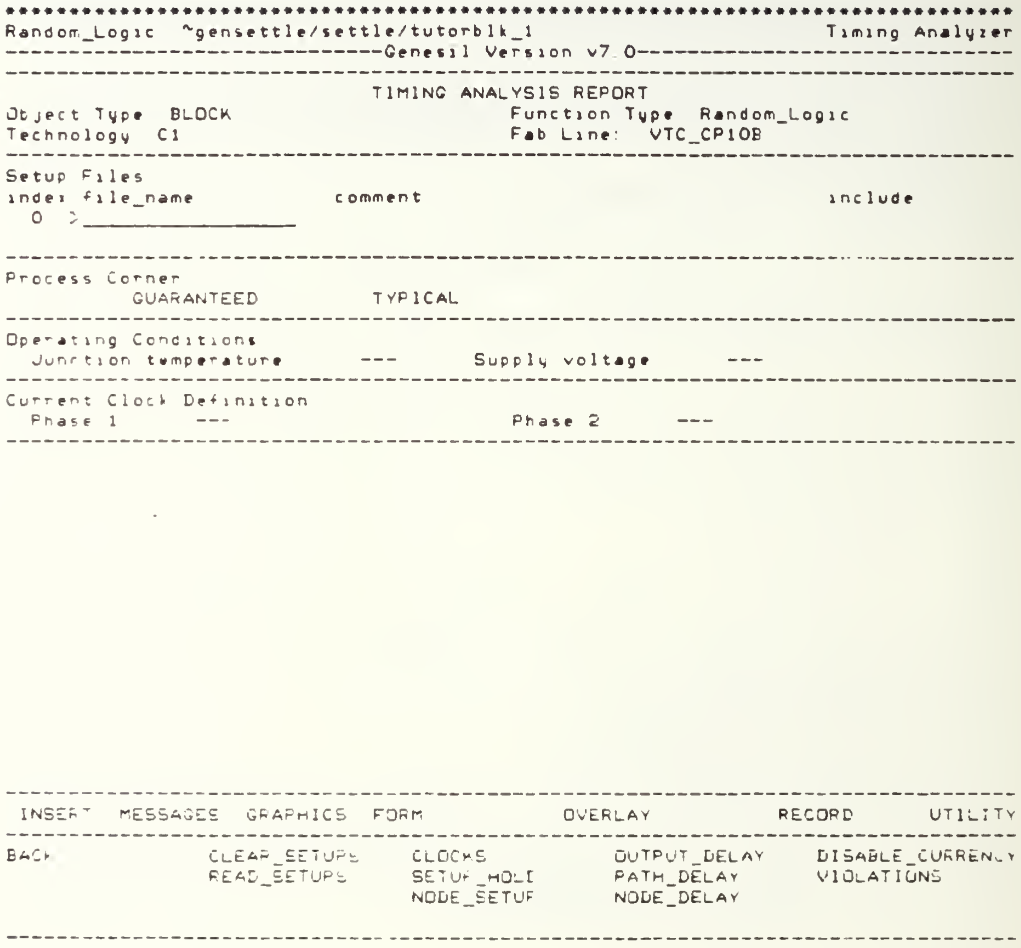
e. Select PATH_DELAY (Figure A.23). The screen now shows a list of all user generated nodes or signals (Figure A.25). By selecting source and destination signals from the list, the system calculates logic propagation delays between the selected nodes (Figure A.26). This is where the detailed schematic may be useful.

f. Select BACK (Figure A.26) to get to Figure A.23, then BACK to exit timing, followed by CONFIRM.



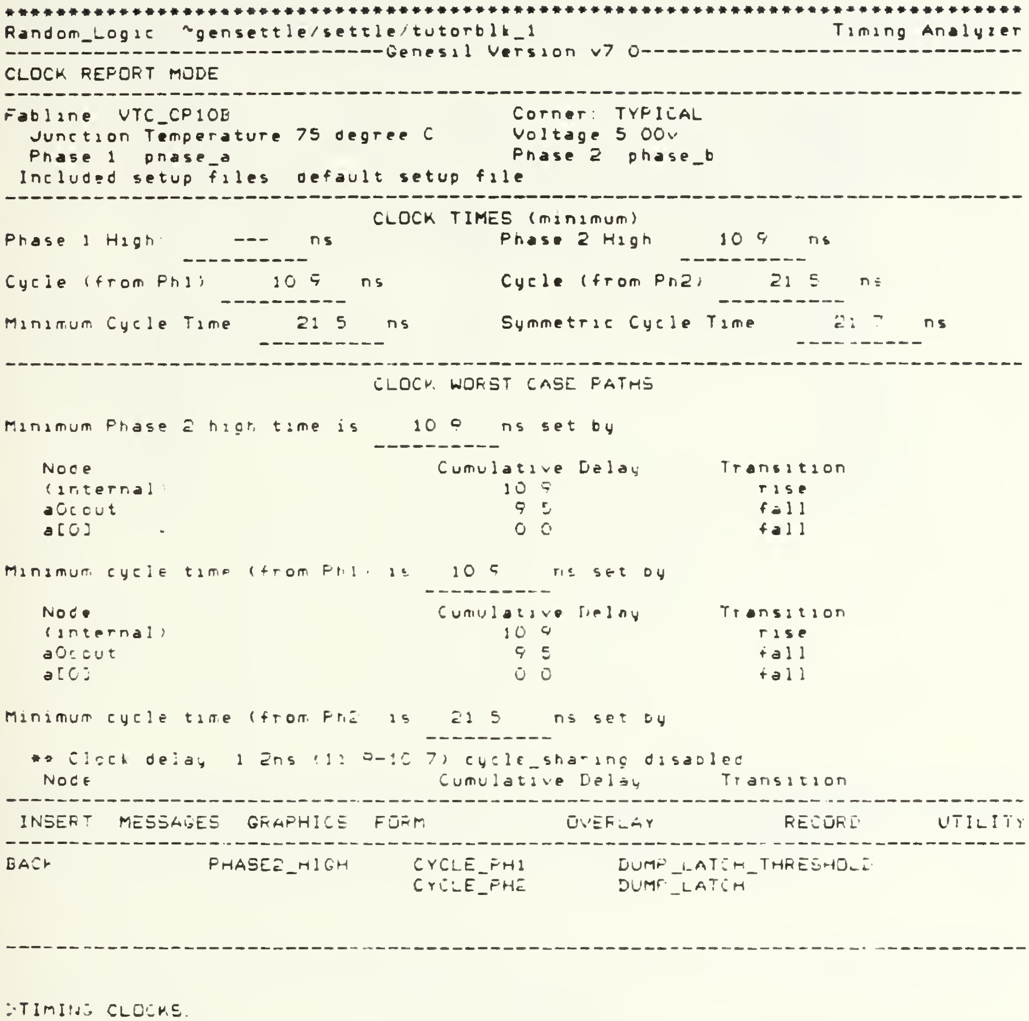
```
*******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                   Functional Simulator
-----------------------------Genesil Version v7.0-------------------------------
)  loading model for type RL
)  Generating model for /mnt/gen/gensettle/settle/tutorblk_1 type RL
) Done with command  MARK SIMULATOR BLOCK MODEL--------in Block   /tutorblk_1
) Times  real=31s, cpu=11s (u=6.9s, s=4.3s) (c=10s)
)  Done with currency check
)  Linking sim model
) phasea= phase_a  phaseb= phase_b
BIND
MULTIPLE_SIGS
a[0]
0
a[1]
1
a[2]
0
a[3]
1
b[0]
1
b[1]
0
b[2]
1
b[3]
1
BACK
CYCLE
.
p1
) .tutorblk_1  is of type  genblock/rl  with  10  ports
) port 0 CI phase_a  to  NC = 1
) port 1 CI phase_b  to  NC = 0
) port 2 O dff1[4 0]  to  NC*5 = 10111
) port 3 I a[3 0]  to  NC*4 = 1010
) port 4 I b[3 0]  to  NC*4 = 1101
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                   OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
BACK            QUERY           HIER_LEVEL      ENVIRONMENT     NEWSCREENS
BIND            CYCLE           RUN_VECTORS     SCROLL          PICK_SCREEN
ASSERT          STEP            UNBIND                          FORMAT_SCREEN
                PROPAGATE       VERIFY_VALUE
--------------------------------------------------------------------------------

Command
 SIMULATION.
```

Figure A.22



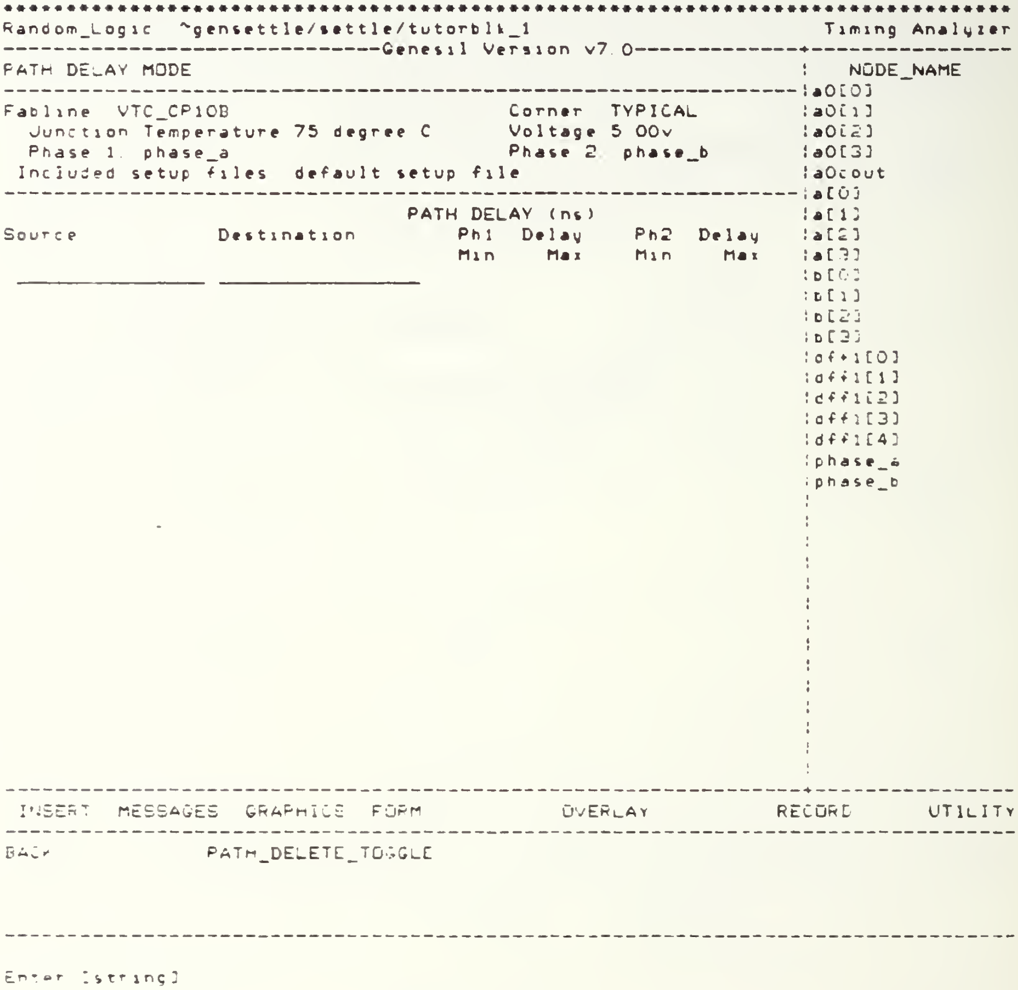
```
*********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                         Timing Analyzer
-----------------------------Genesil Version v7.0-----------------------------
--------------------------------------------------------------------------------
                              TIMING ANALYSIS REPORT
Object Type  BLOCK                         Function Type  Random_Logic
Technology  C1                             Fab Line:  VTC_CP10B
--------------------------------------------------------------------------------
Setup Files
index file_name            comment                                   include
  0  >________________

--------------------------------------------------------------------------------
Process Corner
          GUARANTEED          TYPICAL
--------------------------------------------------------------------------------
Operating Conditions
  Junction temperature      ---      Supply voltage      ---
--------------------------------------------------------------------------------
Current Clock Definition
  Phase 1       ---                      Phase 2      ---
--------------------------------------------------------------------------------







--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
BACK             CLEAR_SETUPS    CLOCKS          OUTPUT_DELAY    DISABLE_CURRENCY
                 READ_SETUPS     SETUP_HOLD      PATH_DELAY      VIOLATIONS
                                 NODE_SETUP      NODE_DELAY

--------------------------------------------------------------------------------

:TIMING:
```

Figure A.23



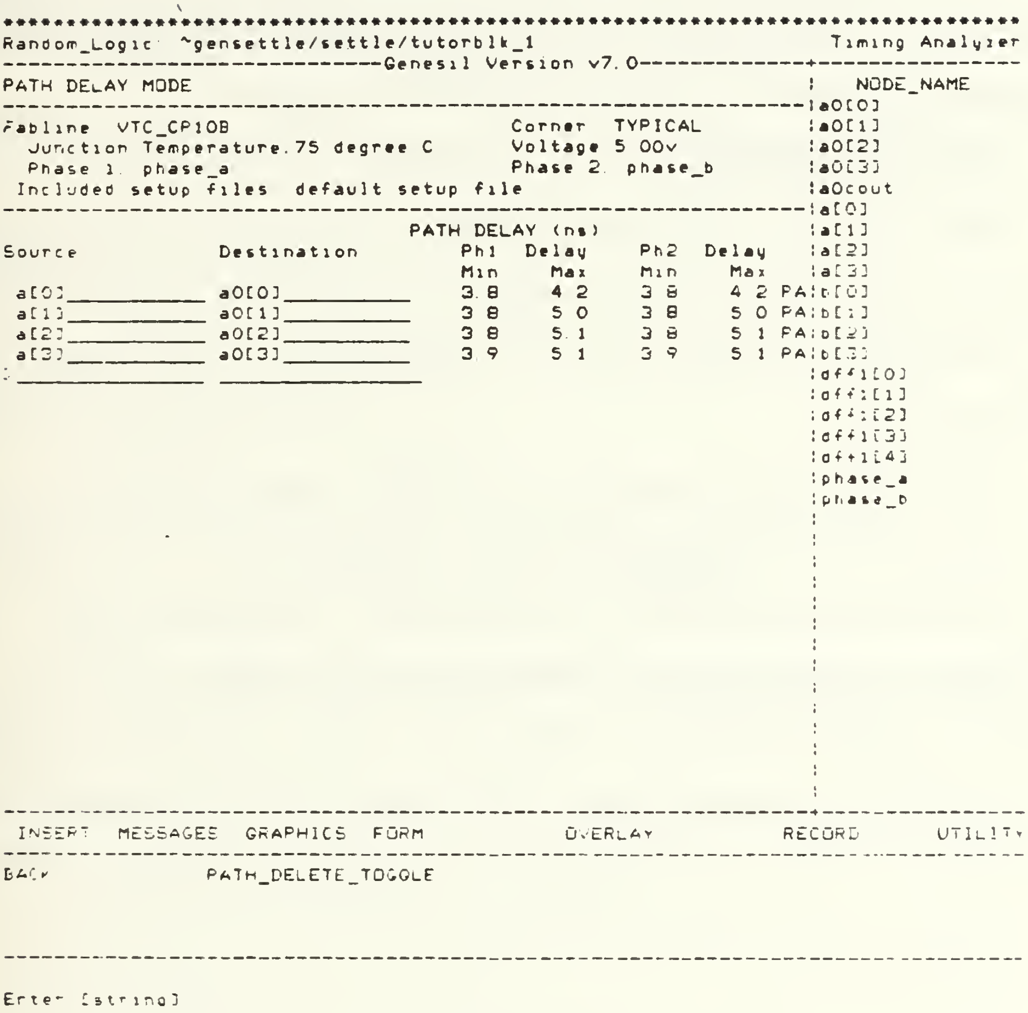
```
*****************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                   Timing Analyzer
------------------------------Genesil Version v7 0---------------------------
CLOCK REPORT MODE
-----------------------------------------------------------------------------
Fabline  VTC_CP10B                        Corner: TYPICAL
  Junction Temperature 75 degree C        Voltage 5 00v
  Phase 1  phase_a                        Phase 2  phase_b
 Included setup files  default setup file
-----------------------------------------------------------------------------
                             CLOCK TIMES (minimum)
Phase 1 High       ---    ns             Phase 2 High      10 9    ns
               ----------                              ----------
Cycle (from Ph1)      10 9    ns         Cycle (from Ph2)      21 5    ns
                  ----------                              ----------
Minimum Cycle Time      21 5    ns       Symmetric Cycle Time      21 7    ns
                     ----------                               ----------
-----------------------------------------------------------------------------
                             CLOCK WORST CASE PATHS

Minimum Phase 2 high time is     10 9    ns set by
                              ----------
   Node                                Cumulative Delay      Transition
   (internal)                                10 9               rise
   a0cout                                     9 5               fall
   a[0]                                       0 0               fall

Minimum cycle time (from Ph1) is    10 9    ns set by
                                 ----------
   Node                                Cumulative Delay      Transition
   (internal)                                10 9               rise
   a0cout                                     9 5               fall
   a[0]                                       0 0               fall

Minimum cycle time (from Ph2) is    21 5    ns set by
                                 ----------
  ** Clock delay  1 2ns (11 9-10 7) cycle_sharing disabled
   Node                                Cumulative Delay      Transition
-----------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY           RECORD     UTILITY
-----------------------------------------------------------------------------
BACK            PHASE2_HIGH      CYCLE_PH1       DUMP_LATCH_THRESHOLD
                                 CYCLE_PH2       DUMP_LATCH


-----------------------------------------------------------------------------

>TIMING CLOCKS.
```

Figure A.24



```
*********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_1                       Timing Analyzer
------------------------------Genesil Version v7.0--------------+----------------
PATH DELAY MODE                                                 :   NODE_NAME
----------------------------------------------------------------:a0[0]
Fabline  VTC_CP10B                        Corner  TYPICAL       :a0[1]
  Junction Temperature 75 degree C        Voltage 5.00v         :a0[2]
  Phase 1. phase_a                        Phase 2. phase_b      :a0[3]
 Included setup files  default setup file                       :a0cout
----------------------------------------------------------------:a[0]
                                   PATH DELAY (ns)              :a[1]
Source           Destination       Ph1  Delay    Ph2  Delay     :a[2]
                                   Min    Max    Min    Max     :a[3]
 _______________  ________________                              :b[0]
                                                                :b[1]
                                                                :b[2]
                                                                :b[3]
                                                                :dff1[0]
                                                                :dff1[1]
                                                                :dff1[2]
                                                                :dff1[3]
                                                                :dff1[4]
                                                                :phase_a
                                                                :phase_b
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
                                                                :
---------------------------------------------------------------+-----------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD       UTILITY
---------------------------------------------------------------------------------
BACK            PATH_DELETE_TOGGLE


---------------------------------------------------------------------------------

Enter [string]
 TIMING: PATH_DELAY>
```

Figure A.25



```
******************************************************************************
Random_Logic: ~gensettle/settle/tutorblk_1                      Timing Analyzer
----------------------------Genesil Version v7.0--------------+----------------
PATH DELAY MODE                                               !   NODE_NAME
--------------------------------------------------------------!a0[0]
Fabline  VTC_CP10B                      Corner  TYPICAL       !a0[1]
  Junction Temperature.75 degree C      Voltage 5 00v         !a0[2]
  Phase 1. phase_a                      Phase 2. phase_b      !a0[3]
 Included setup files  default setup file                     !a0cout
--------------------------------------------------------------!a[0]
                               PATH DELAY (ns)                !a[1]
Source           Destination        Ph1  Delay     Ph2  Delay  !a[2]
                                    Min    Max     Min    Max  !a[3]
 a[0]__________  a0[0]__________    3.8    4 2     3 8    4 2 PA!b[0]
 a[1]__________  a0[1]__________    3 8    5 0     3 8    5 0 PA!b[1]
 a[2]__________  a0[2]__________    3 8    5.1     3 8    5 1 PA!b[2]
 a[3]__________  a0[3]__________    3.9    5 1     3 9    5 1 PA!b[3]
: ______________ ________________                              !dff1[0]
                                                               !dff1[1]
                                                               !dff1[2]
                                                               !dff1[3]
                                                               !dff1[4]
                                                               !phase_a
                                                               !phase_b
                                                               !
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
---------------------------------------------------------------+---------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD       UTILITY
-------------------------------------------------------------------------------
BACK            PATH_DELETE_TOGGLE


-------------------------------------------------------------------------------

Enter [string]
[TIMING:PATH_DELAY]
```

Figure A.26

6. Now to exit Genesil:

   a. Select EXIT_GENESIL (Figure A.15).

   b. Select CONFIRM.

   c. Select appopriate log command. The block is stored in the user's account regardless of which log command is selected. It is best to not save the log in the interest of memory, unless a future printout is desired.

C. TUTORBLK_2 BLOCK

This section is a user exercise to build a block named tutorblk_2 by following the steps illustrated in section B. This block is necessary for the completion of the chip in section E.

The block will contain six random logic objects, consisting of 5 inverters (i0-i4) and a 5 bit D F/F (DFF5). The pre-planned schematic of the block, including all necessary signal names is shown in Figure A.27.

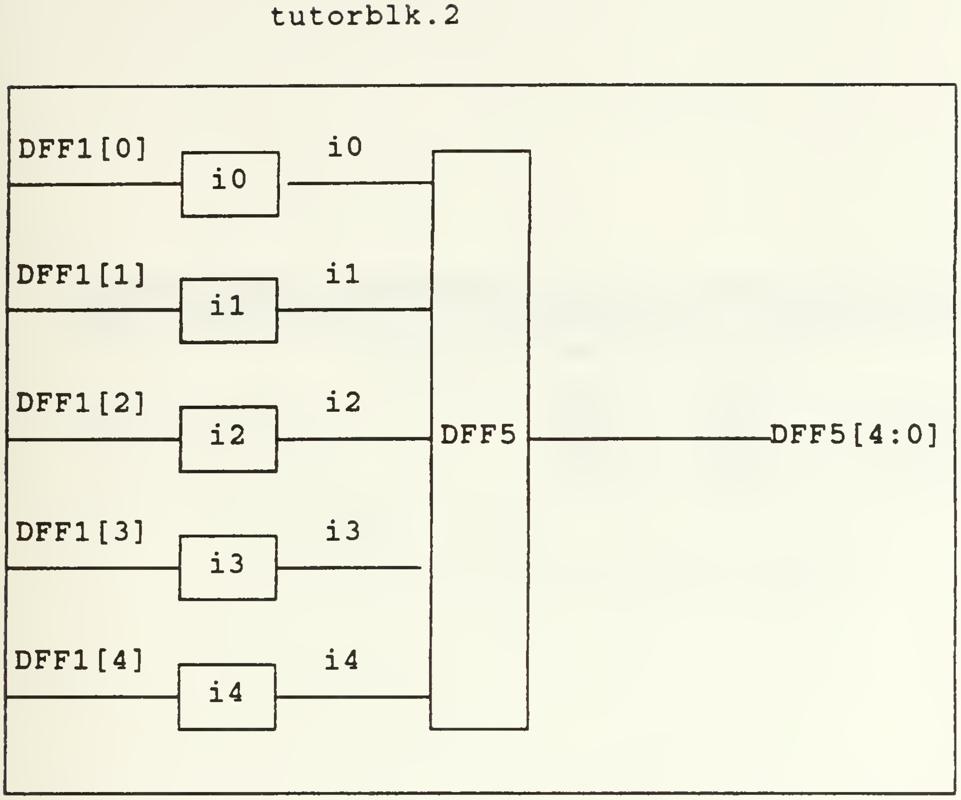
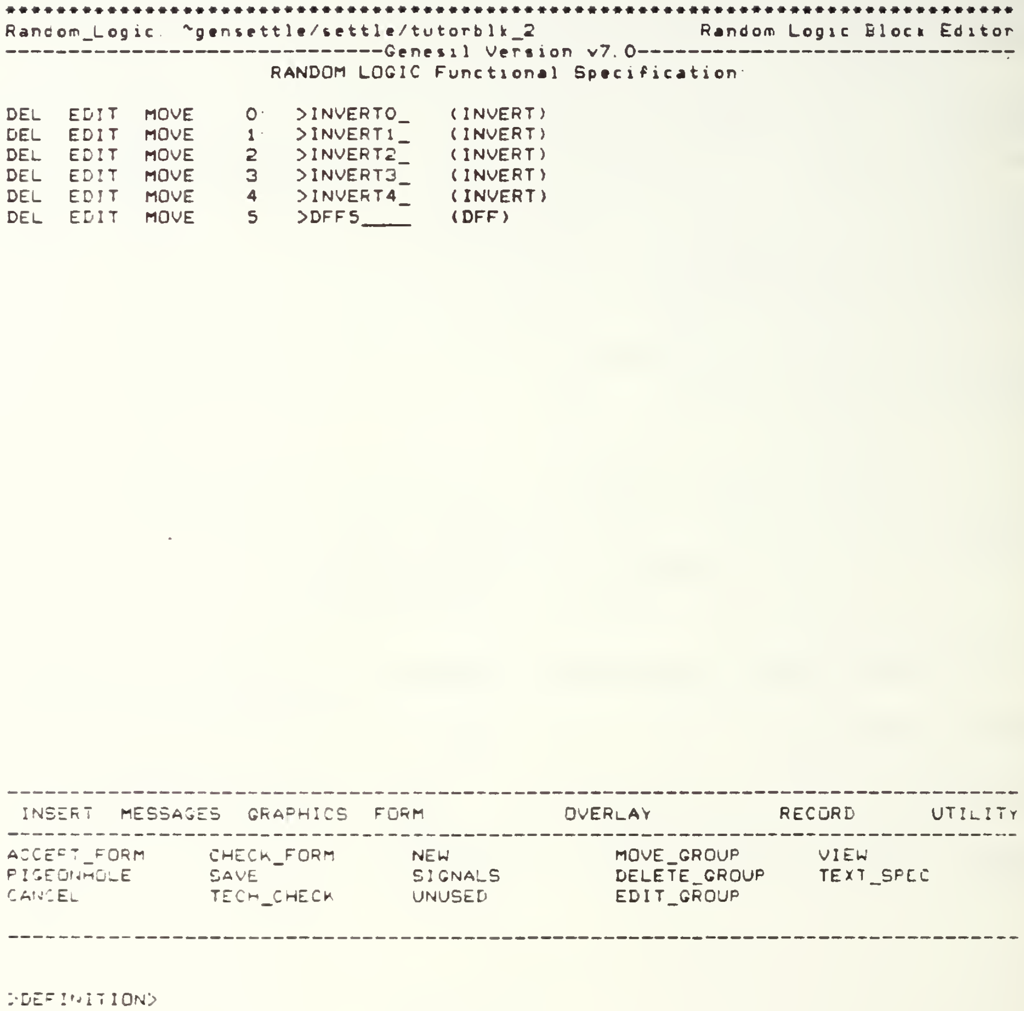


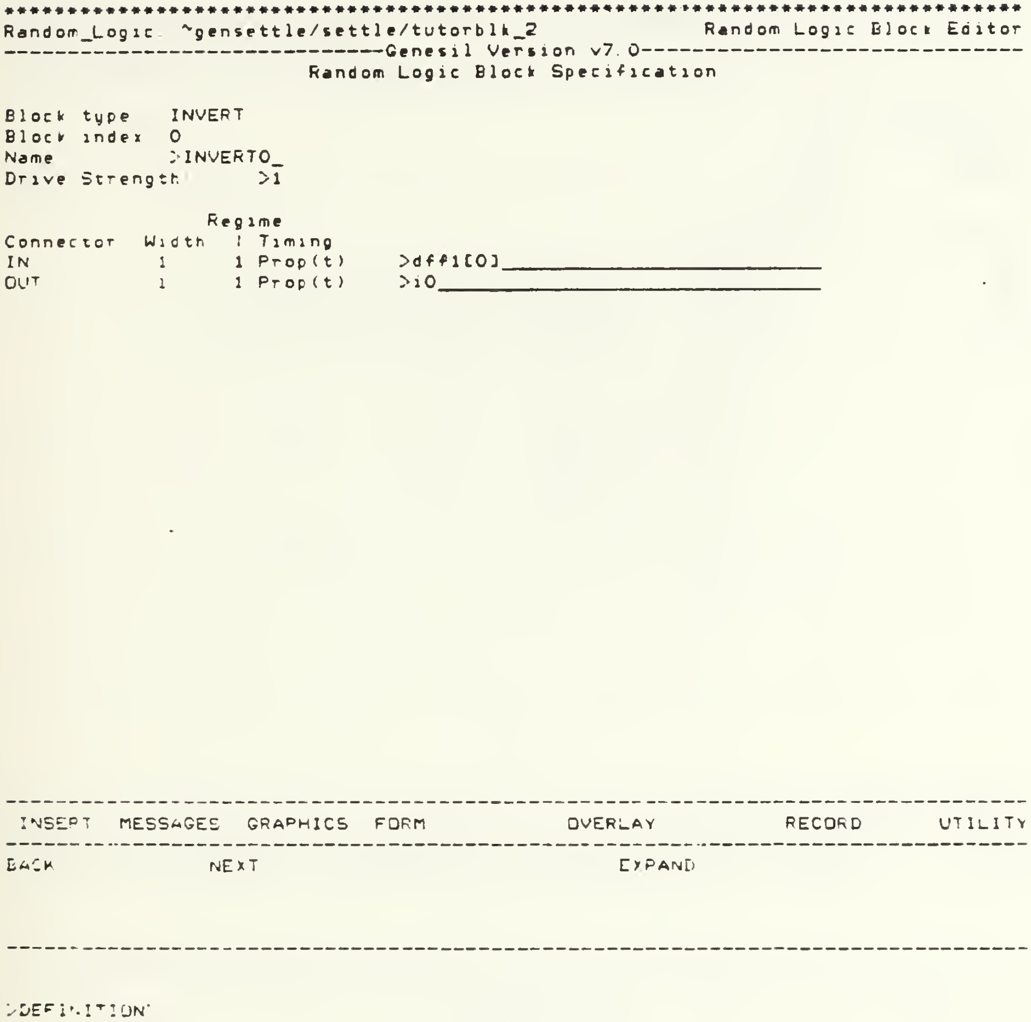
Figure A.27 Tutorblk_2

To ensure that the block functions properly and will connect properly to the chip, make the specification menus match Figure A.28 through A.32. Then compile, simulate, and perform timing analysis as in section B.



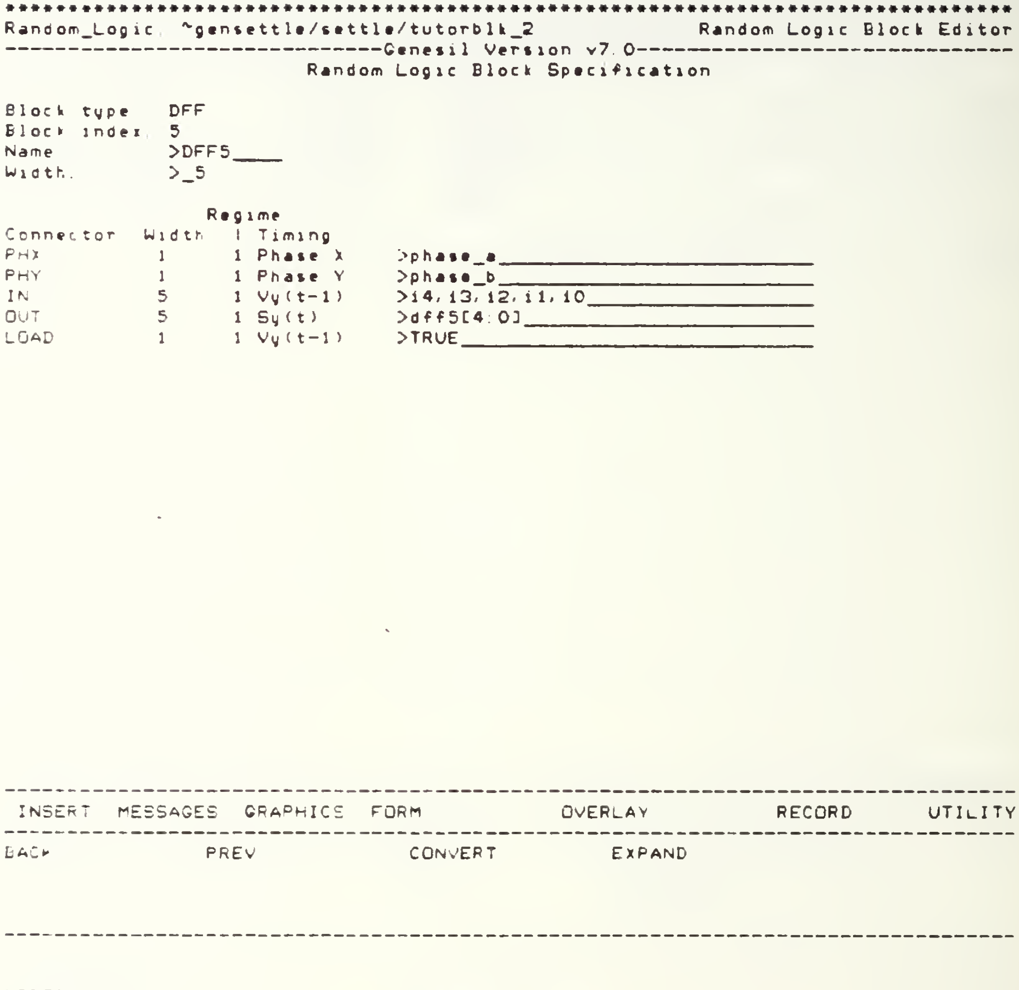
```
********************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_2           Random Logic Block Editor
-----------------------------Genesil Version v7.0-------------------------------
                    RANDOM LOGIC Functional Specification

DEL   EDIT   MOVE     0   >INVERT0_    (INVERT)
DEL   EDIT   MOVE     1   >INVERT1_    (INVERT)
DEL   EDIT   MOVE     2   >INVERT2_    (INVERT)
DEL   EDIT   MOVE     3   >INVERT3_    (INVERT)
DEL   EDIT   MOVE     4   >INVERT4_    (INVERT)
DEL   EDIT   MOVE     5   >DFF5____    (DFF)















--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_FORM      NEW             MOVE_GROUP      VIEW
PIGEONHOLE      SAVE            SIGNALS         DELETE_GROUP    TEXT_SPEC
CANCEL          TECH_CHECK      UNUSED          EDIT_GROUP

--------------------------------------------------------------------------------

:DEFINITION>
```

Figure A.28



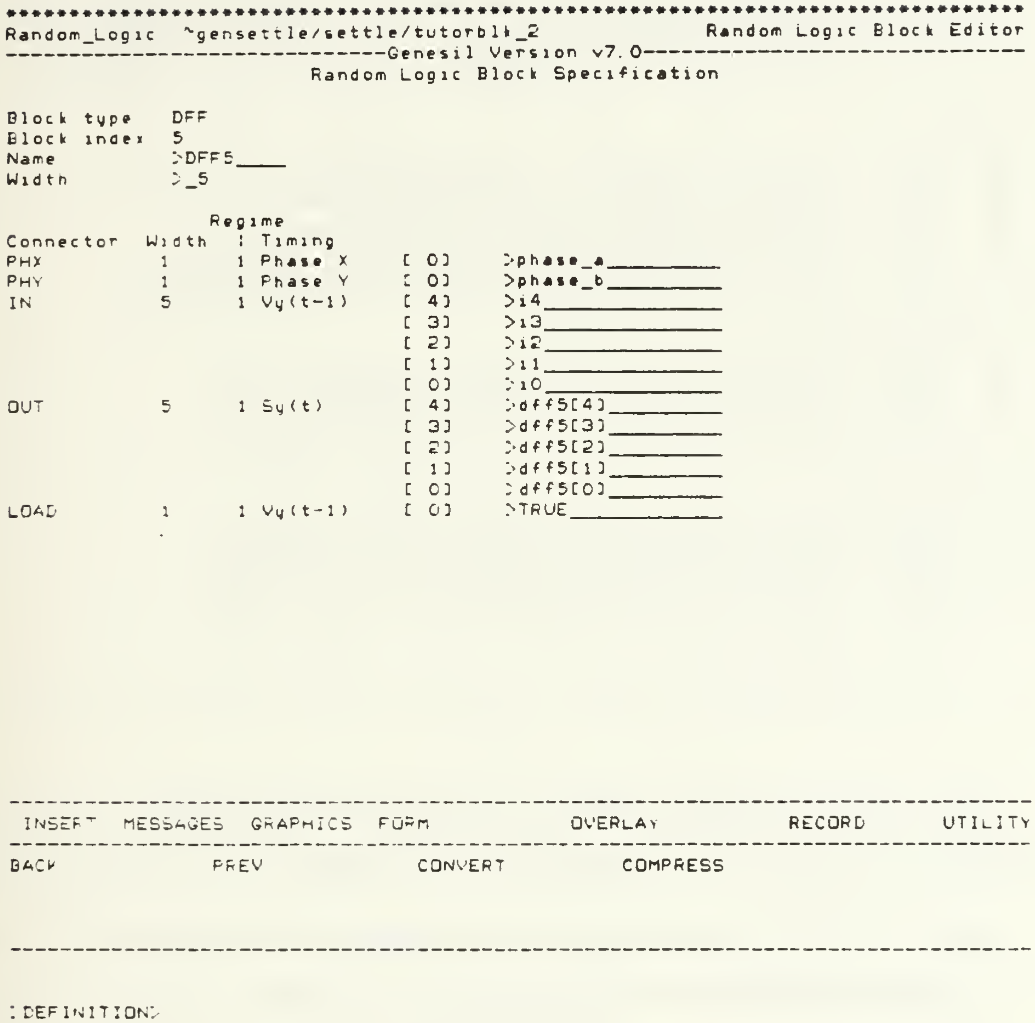
```
*******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_2            Random Logic Block Editor
------------------------------Genesil Version v7.0-----------------------------
                        Random Logic Block Specification

Block type    INVERT
Block index   0
Name         >INVERT0_
Drive Strength      >1

                 Regime
Connector  Width  | Timing
IN           1      1 Prop(t)     >dff1[0]_________________________
OUT          1      1 Prop(t)     >i0______________________________










-------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
-------------------------------------------------------------------------------
BACK            NEXT                             EXPAND



-------------------------------------------------------------------------------


>DEFINITION
```

Figure A.29



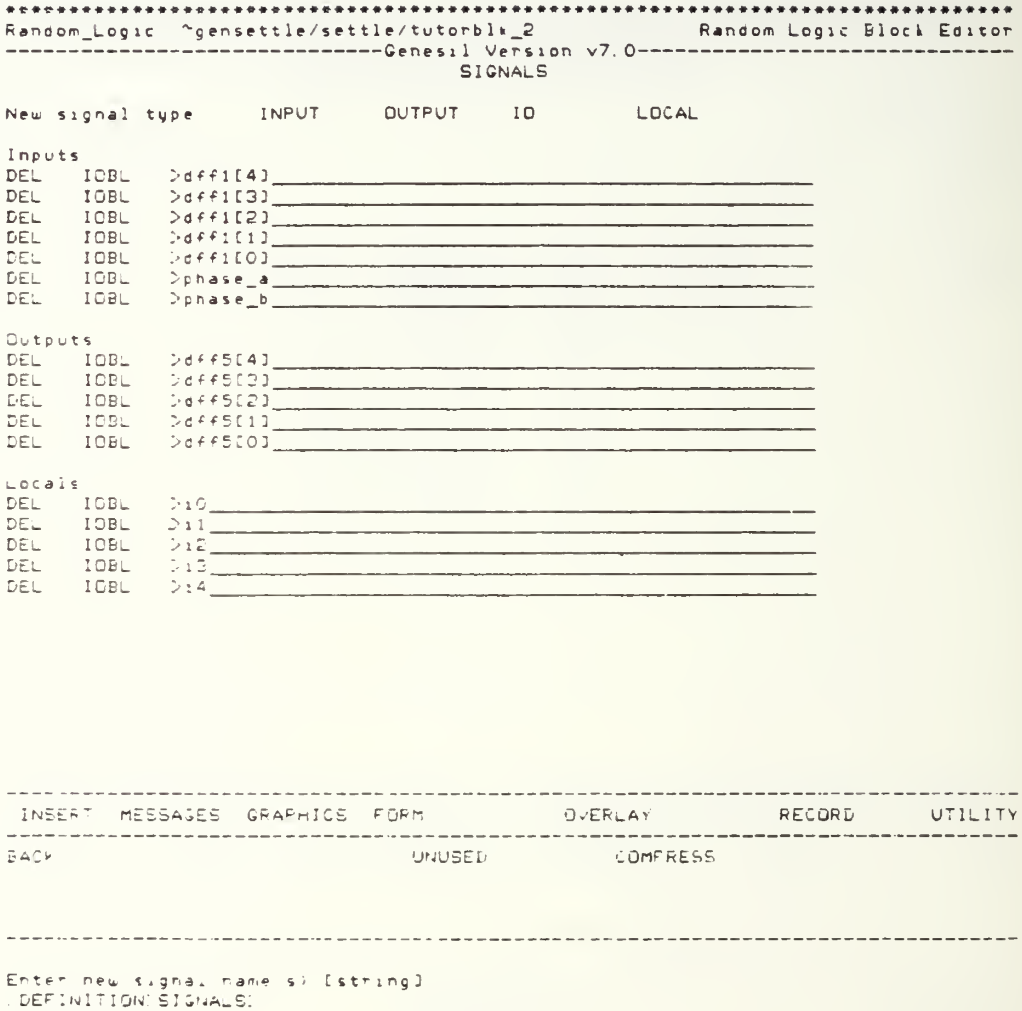
```
******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_2                Random Logic Block Editor
------------------------------Genesil Version v7.0-------------------------------
                              Random Logic Block Specification

Block type    DFF
Block index   5
Name          >DFF5____
Width         >_5

                 Regime
Connector  Width   I  Timing
PHX          1     1  Phase X     >phase_a__________________________
PHY          1     1  Phase Y     >phase_b__________________________
IN           5     1  Vy(t-1)     >i4,i3,i2,i1,i0___________________
OUT          5     1  Sy(t)       >dff5[4:0]________________________
LOAD         1     1  Vy(t-1)     >TRUE_____________________________










---------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD        UTILITY
---------------------------------------------------------------------------------
BACK            PREV            CONVERT         EXPAND



---------------------------------------------------------------------------------

>DEFINITION>
```

Figure A.30



```
**************************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_2                   Random Logic Block Editor
--------------------------------Genesil Version v7.0-----------------------------------
                                Random Logic Block Specification

Block type    DFF
Block index   5
Name          >DFF5____
Width         >_5

                    Regime
Connector  Width   : Timing
PHX          1       1 Phase X      [ 0]      >phase_a__________
PHY          1       1 Phase Y      [ 0]      >phase_b__________
IN           5       1 Vy(t-1)      [ 4]      >i4_______________
                                    [ 3]      >i3_______________
                                    [ 2]      >i2_______________
                                    [ 1]      >i1_______________
                                    [ 0]      >i0_______________
OUT          5       1 Sy(t)        [ 4]      >dff5[4]__________
                                    [ 3]      >dff5[3]__________
                                    [ 2]      >dff5[2]__________
                                    [ 1]      >dff5[1]__________
                                    [ 0]      >dff5[0]__________
LOAD         1       1 Vy(t-1)      [ 0]      >TRUE_____________





-----------------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY            RECORD          UTILITY
-----------------------------------------------------------------------------------------
BACK            PREV             CONVERT           COMPRESS



-----------------------------------------------------------------------------------------

:DEFINITION>
```

Figure A.31



```
******************************************************************************
Random_Logic  ~gensettle/settle/tutorblk_2             Random Logic Block Editor
------------------------------Genesil Version v7.0------------------------------
                                   SIGNALS

New signal type      INPUT      OUTPUT     IO         LOCAL

Inputs
DEL     IOBL    >dff1[4]__________________________________________
DEL     IOBL    >dff1[3]__________________________________________
DEL     IOBL    >dff1[2]__________________________________________
DEL     IOBL    >dff1[1]__________________________________________
DEL     IOBL    >dff1[0]__________________________________________
DEL     IOBL    >phase_a__________________________________________
DEL     IOBL    >phase_b__________________________________________

Outputs
DEL     IOBL    >dff5[4]__________________________________________
DEL     IOBL    >dff5[3]__________________________________________
DEL     IOBL    >dff5[2]__________________________________________
DEL     IOBL    >dff5[1]__________________________________________
DEL     IOBL    >dff5[0]__________________________________________

Locals
DEL     IOBL    >i0_______________________________________________
DEL     IOBL    >i1_______________________________________________
DEL     IOBL    >i2_______________________________________________
DEL     IOBL    >i3_______________________________________________
DEL     IOBL    >i4_______________________________________________




--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY         RECORD        UTILITY
--------------------------------------------------------------------------------
BACK                                UNUSED          COMPRESS



--------------------------------------------------------------------------------

Enter new signal name(s) [string]
(DEFINITION SIGNALS)
```

Figure A.32

D. MULT_MOD MODULE

This section illustrates the design of a 4 bit multiplier module. The detailed schematic is shown in Figure A.33.

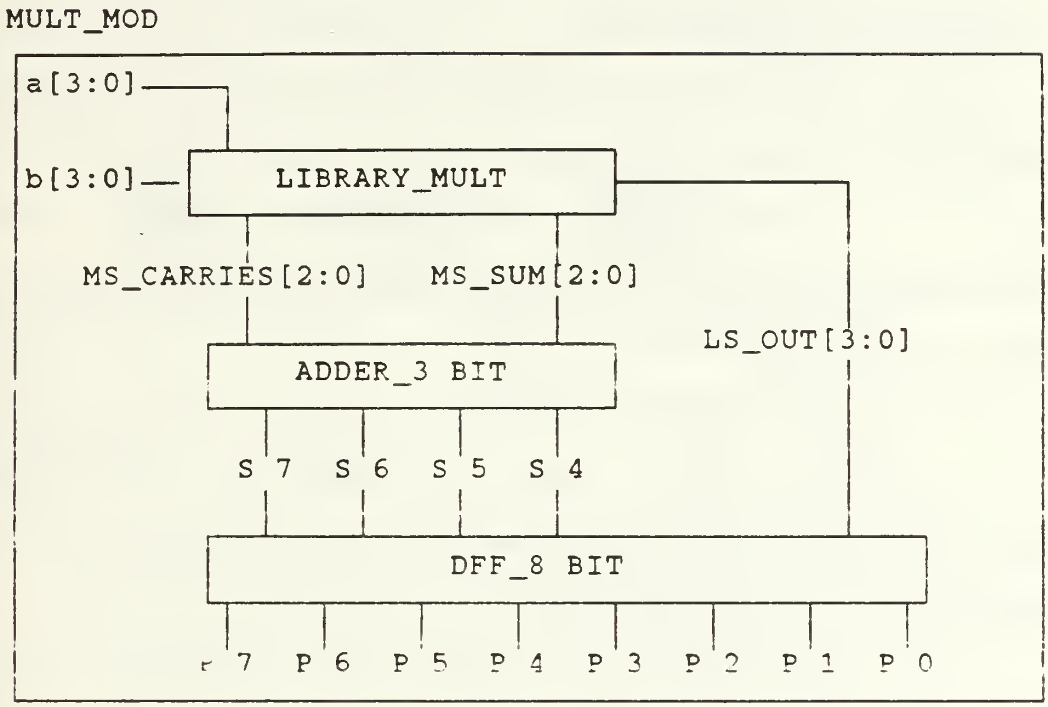


Figure A.33 Mult_Mod Module

Library_mult is a Genesil system library parallel multiplier block. The external adder is necessary to complete the multiplication. Detailed information concerning the multiplier block is contained in the Genesil Silicon Compiler Library Volume I, Blocks.

1. The user should proceed as in the previous sections up through ATTACH NEW (Figure A.5):

   a. Now select GENERAL_MODULE (Figure A.5).

b. Go BACK, name the module Mult_Mod, by typing it in next to the prompt, and attach it to the tree as in the previous sections.

c. Select DEFINITION from the Executive menu.

d. Select HEADER (Figure A.34), then VTC_CP10B for Fab line, and ACCEPT_FORM.

e. Now select SPECIFICATION to pull up the Module specification menu. No objects are on the menu yet.

f. Select ATTACH_NEW, then RANDOM_LOGIC. The user will now be prompted for a name. Name the first random logic object adder_3bit.

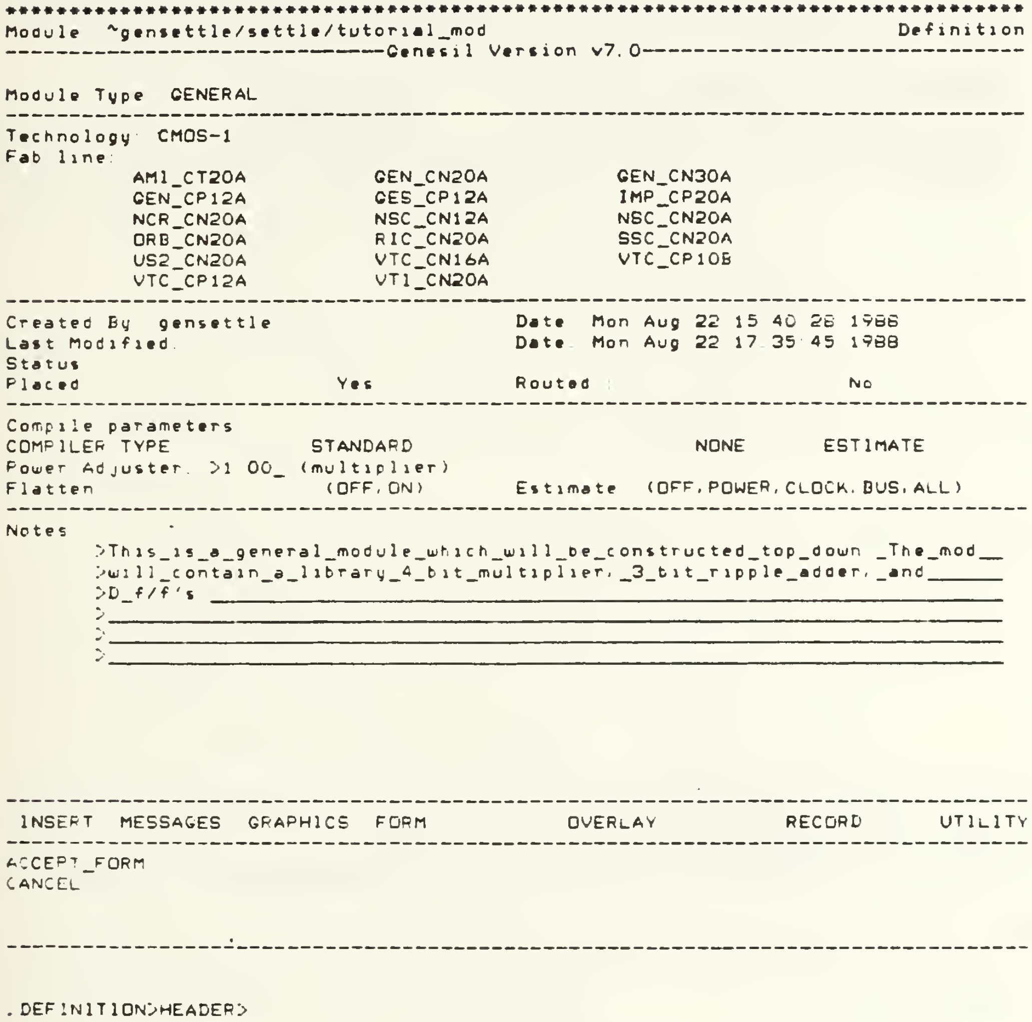
g. Repeat for dff_8bit.

h. Now select ATTACH_NEW, then select BLOCK, and name it library_mult when prompted. When complete, the screen should look like Figure A.35.

i. Each sub_object (adder_3bit, dff_8bit, and library_mult) must now be defined.

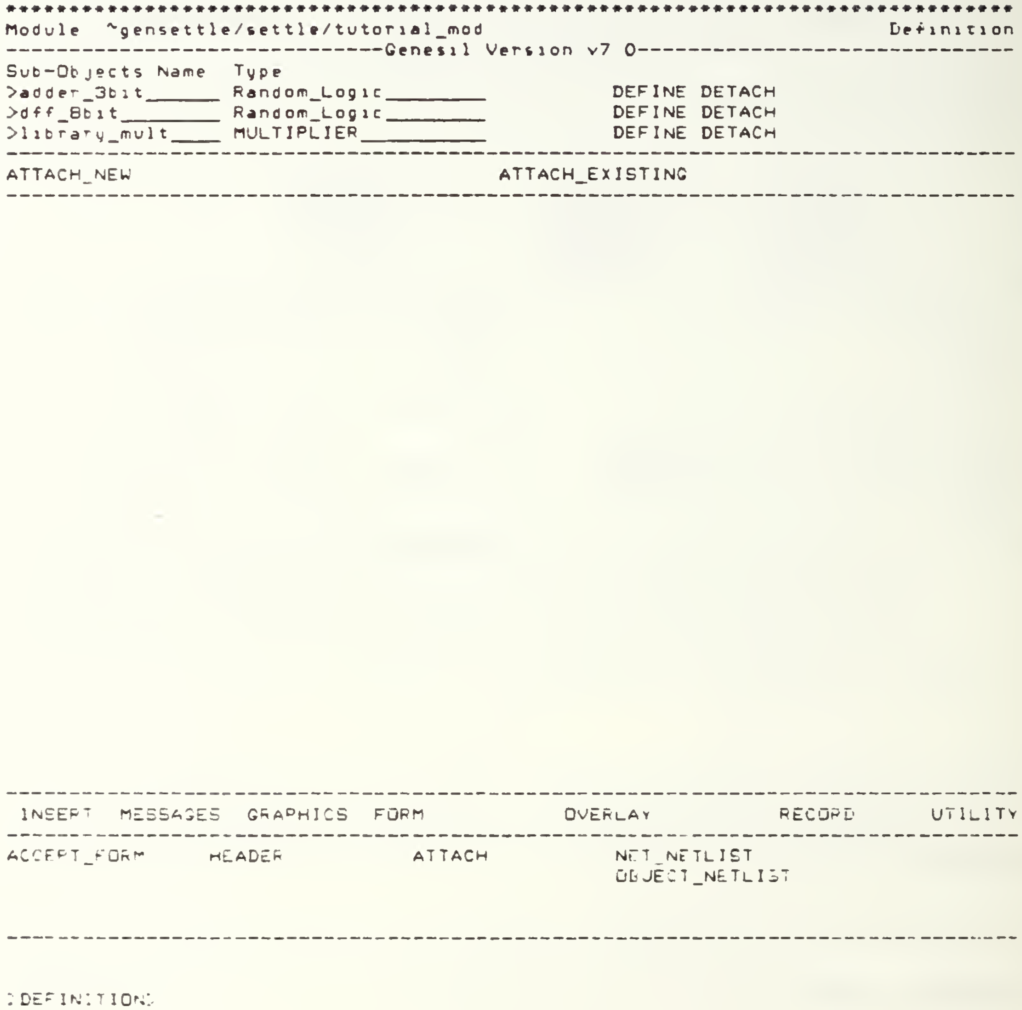
j. Starting with adder_3bit, select DEFINE (Figure A.35). Then select HEADER and specify RANDOM_LOGIC for object type on the Header form. Fab line can be specified, but will be automatically taken care of at the module level. Select ACCEPT_FORM.

k. Next select SPECIFICATION which pulls up a RANDOM LOGIC Functional Specification Menu. Select NEW, then ADDER from the menu provided.



```
*******************************************************************************
Module  ~gensettle/settle/tutorial_mod                                Definition
------------------------------Genesil Version v7.0------------------------------

Module Type  GENERAL
--------------------------------------------------------------------------------
Technology  CMOS-1
Fab line:
          AMI_CT20A            GEN_CN20A            GEN_CN30A
          GEN_CP12A            GES_CP12A            IMP_CP20A
          NCR_CN20A            NSC_CN12A            NSC_CN20A
          ORB_CN20A            RIC_CN20A            SSC_CN20A
          US2_CN20A            VTC_CN16A            VTC_CP10B
          VTC_CP12A            VTI_CN20A
--------------------------------------------------------------------------------
Created By  gensettle                   Date  Mon Aug 22 15 40 28 1988
Last Modified                           Date  Mon Aug 22 17 35 45 1988
Status
Placed                      Yes         Routed                    No
--------------------------------------------------------------------------------
Compile parameters
COMPILER TYPE          STANDARD                          NONE      ESTIMATE
Power Adjuster  >1 00_ (multiplier)
Flatten                 (OFF, ON)       Estimate  (OFF, POWER, CLOCK, BUS, ALL)
--------------------------------------------------------------------------------
Notes
       >This_is_a_general_module_which_will_be_constructed_top_down _The_mod__
       >will_contain_a_library_4_bit_multiplier, _3_bit_ripple_adder, _and______
       >D_f/f's _______________________________________________________________
       >______________________________________________________________________
       >______________________________________________________________________
       >______________________________________________________________________



--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM
CANCEL


--------------------------------------------------------------------------------

.DEFINITION>HEADER>
```

Figure A.34



```
******************************************************************************
Module  ~gensettle/settle/tutorial_mod                             Definition
---------------------------Genesil Version v7 0------------------------------
Sub-Objects Name  Type
>adder_3bit______ Random_Logic________          DEFINE DETACH
>dff_8bit________ Random_Logic________          DEFINE DETACH
>library_mult____ MULTIPLIER__________          DEFINE DETACH
-----------------------------------------------------------------------------
ATTACH_NEW                             ATTACH_EXISTING
-----------------------------------------------------------------------------









-----------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD     UTILITY
-----------------------------------------------------------------------------
ACCEPT_FORM     HEADER          ATTACH          NET_NETLIST
                                                OBJECT_NETLIST


-----------------------------------------------------------------------------

<DEFINITION>
```

Figure A.35

l. Go BACK to the Specification menu and it should look like Figure A.36.

m. Now select EDIT, and fill out the Random Logic Block Specification menu as shown in Figure A.37. Select EXPAND and the menu should look like Figure A.38.

n. Starting at step j, follow the same procedures with dff_8bit. The RANDOM LOGIC Functional Specification form (Figure A.39) and Random Logic Block Specification form (Figure A.40) should be filled out as shown.

o. Proceed the same way for library_mult, but remember to select multiplier on the Header form.

p. Fill out the MULTIPLIER SPECIFICATION menu as shown in Figure A.41.

q. Select ACCEPT_FORM (Figure A.41), then BACK to the Module Specification Form (Figure A.35).

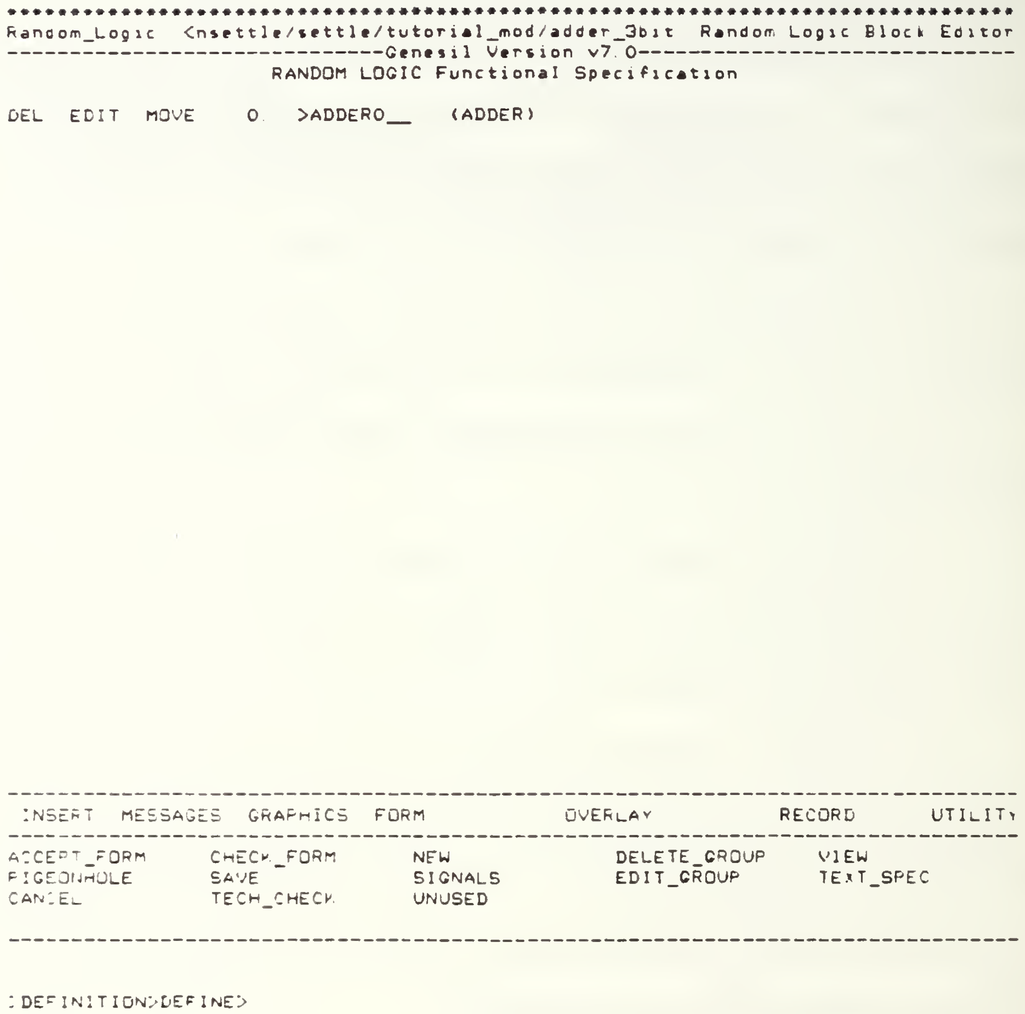
2. The module must now be netlisted:

a. Select OBJECT_NETLIST (Figure A.35).

b. Type in adder_3bit (Figure A.42) next to Object Name. Another way to do this is to mouse Object Name, depress Return, and a list of sub-Objects is pulled up on the right side of the screen for selection.

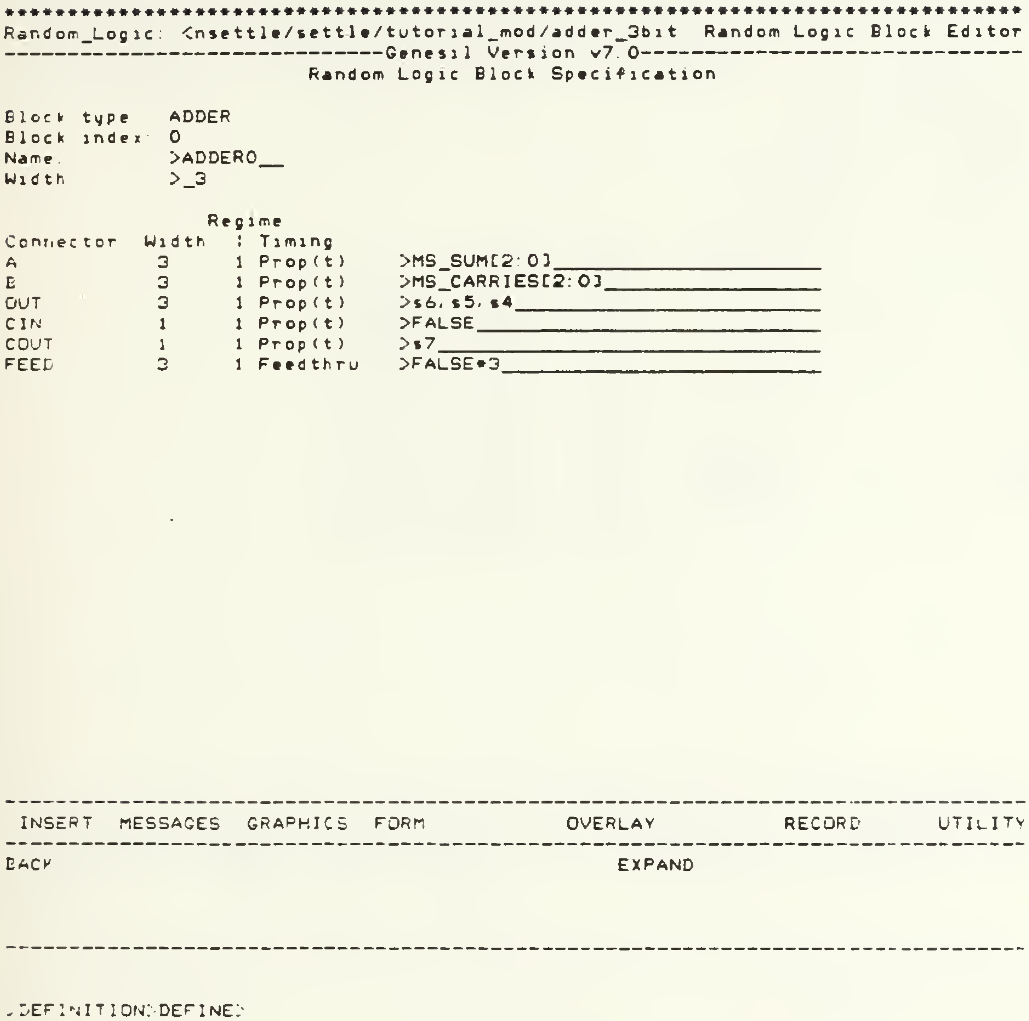
c. Proceed through the Net Name list, and select E, next to attributes, for external for all signals. This is necessary to make the module function correctly!

d. Select Object Name, and type in dff_8bit (Figure A.43). Do the same as in c above.



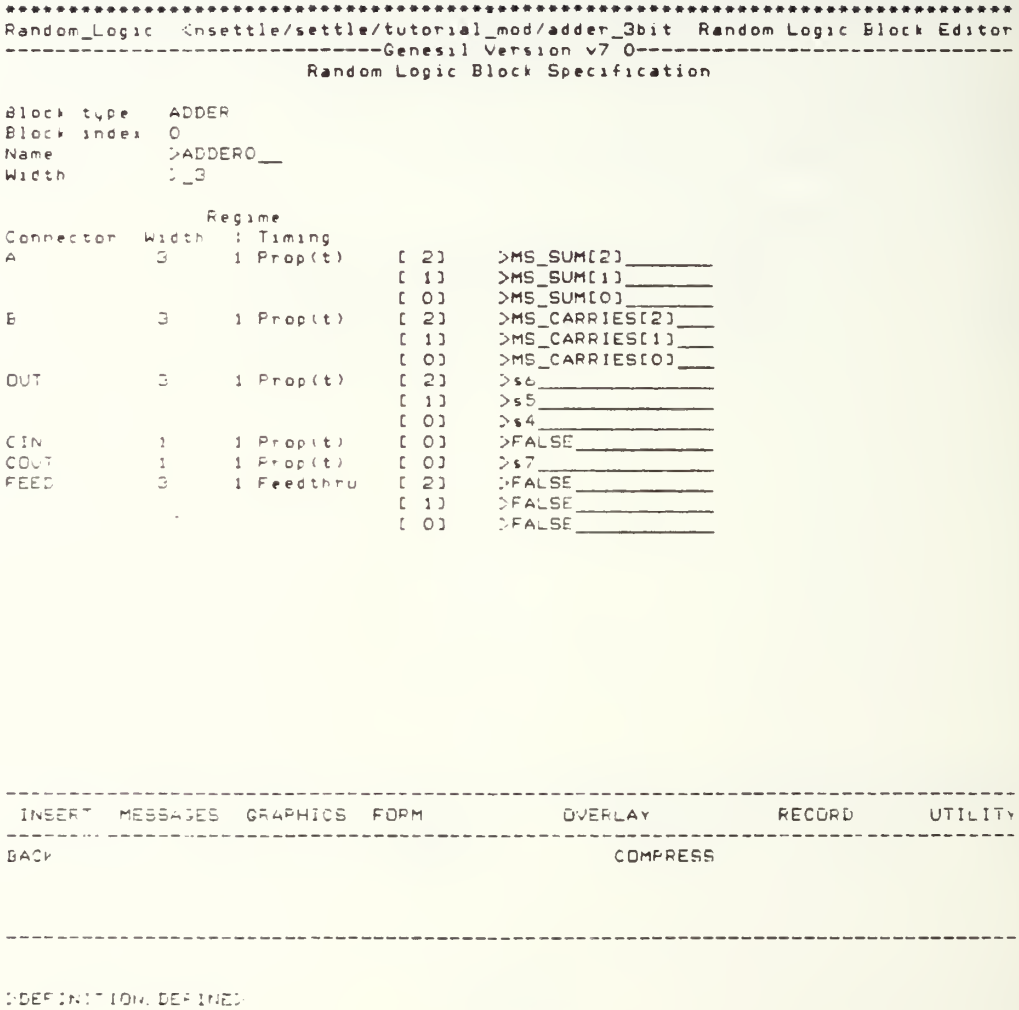
```
*********************************************************************************
Random_Logic  <nsettle/settle/tutorial_mod/adder_3bit  Random Logic Block Editor
------------------------------Genesil Version v7.0-------------------------------
                     RANDOM LOGIC Functional Specification

DEL   EDIT   MOVE     0.   >ADDER0__    (ADDER)


















---------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY           RECORD      UTILITY
---------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_FORM       NEW              DELETE_GROUP     VIEW
PIGEONHOLE       SAVE             SIGNALS          EDIT_GROUP       TEXT_SPEC
CANCEL           TECH_CHECK       UNUSED

---------------------------------------------------------------------------------

:DEFINITION>DEFINE>
```

Figure A.36



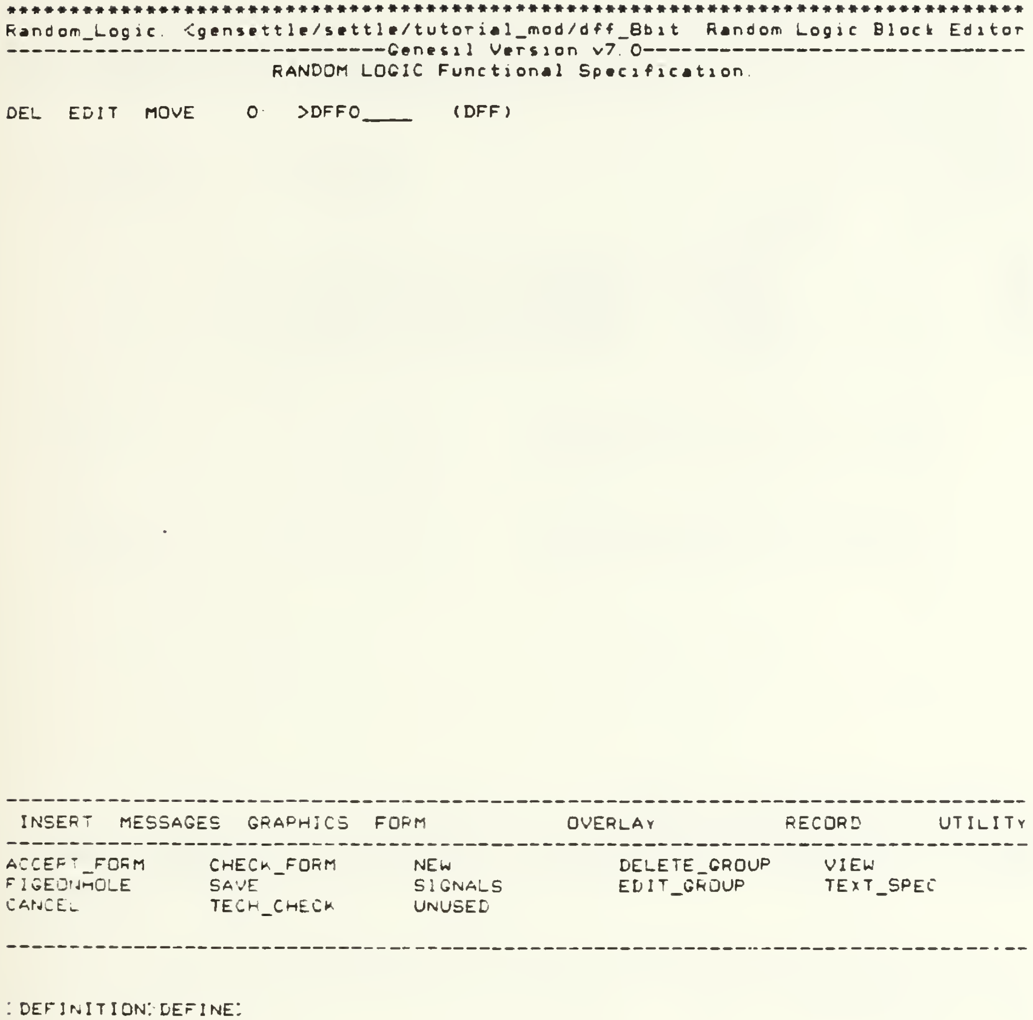
```
******************************************************************************
Random_Logic: <nsettle/settle/tutorial_mod/adder_3bit  Random Logic Block Editor
-----------------------------Genesil Version v7.0-----------------------------
                          Random Logic Block Specification

Block type     ADDER
Block index:   0
Name:          >ADDER0__
Width          >_3

                  Regime
Connector  Width   ! Timing
A            3     1 Prop(t)     >MS_SUM[2:0]____________________
B            3     1 Prop(t)     >MS_CARRIES[2:0]________________
OUT          3     1 Prop(t)     >s6, s5, s4_____________________
CIN          1     1 Prop(t)     >FALSE__________________________
COUT         1     1 Prop(t)     >s7_____________________________
FEED         3     1 Feedthru    >FALSE*3________________________






------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
BACK                                           EXPAND


------------------------------------------------------------------------------

<DEFINITION>DEFINE>
```

Figure A.37



```
********************************************************************************
Random_Logic  <nsettle/settle/tutorial_mod/adder_3bit  Random Logic Block Editor
-----------------------------Genesil Version v7 0-------------------------------
                        Random Logic Block Specification

Block type    ADDER
Block index   0
Name          >ADDER0__
Width         >_3

                  Regime
Connector   Width  : Timing
A             3      1 Prop(t)     [ 2]     >MS_SUM[2]________
                                   [ 1]     >MS_SUM[1]________
                                   [ 0]     >MS_SUM[0]________
B             3      1 Prop(t)     [ 2]     >MS_CARRIES[2]___
                                   [ 1]     >MS_CARRIES[1]___
                                   [ 0]     >MS_CARRIES[0]___
OUT           3      1 Prop(t)     [ 2]     >s6______________
                                   [ 1]     >s5______________
                                   [ 0]     >s4______________
CIN           1      1 Prop(t)     [ 0]     >FALSE___________
COUT          1      1 Prop(t)     [ 0]     >s7______________
FEED          3      1 Feedthru    [ 2]     >FALSE___________
                                   [ 1]     >FALSE___________
                                   [ 0]     >FALSE___________









--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD       UTILITY
--------------------------------------------------------------------------------
BACK                                            COMPRESS



--------------------------------------------------------------------------------


>DEFINITION. DEFINED>
```

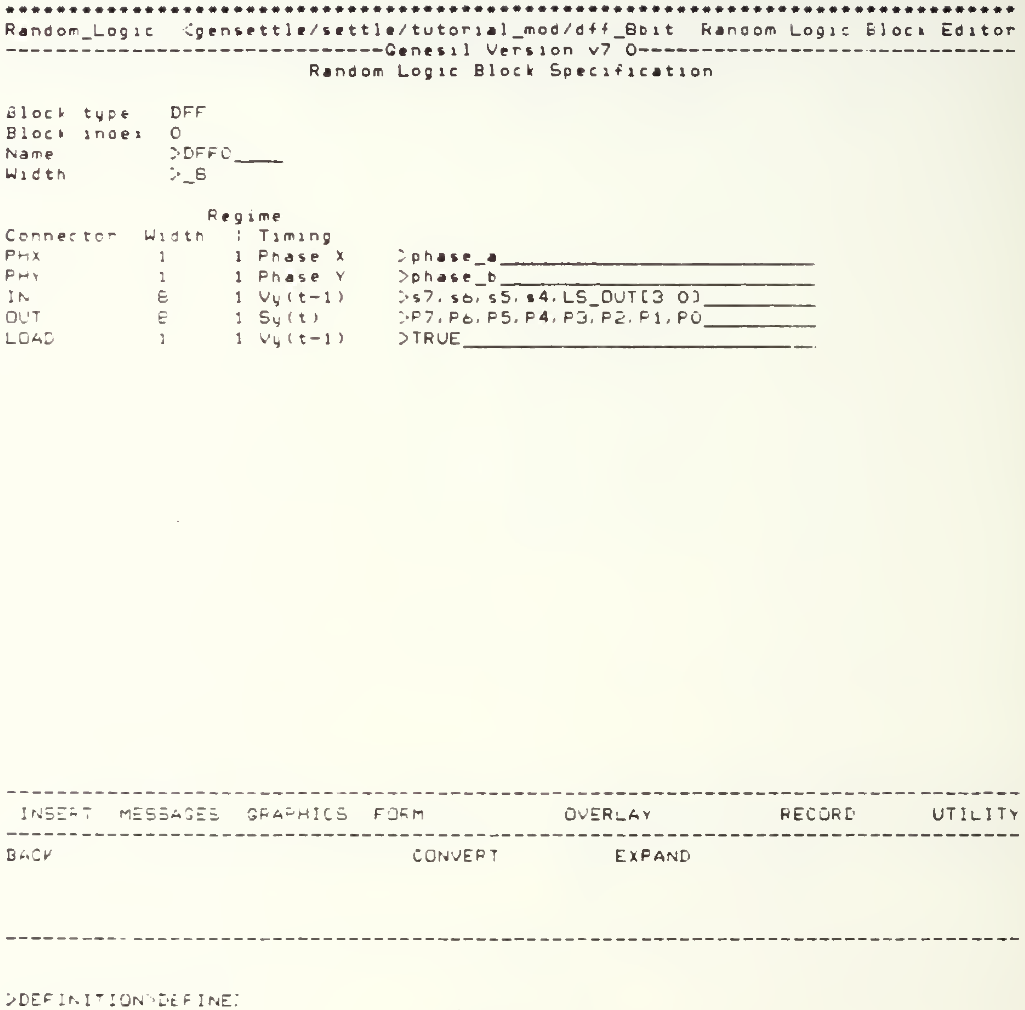
Figure A.38



```
*******************************************************************************
Random_Logic. <gensettle/settle/tutorial_mod/dff_8bit  Random Logic Block Editor
-----------------------------Genesil Version v7.0-------------------------------
                     RANDOM LOGIC Functional Specification.

DEL   EDIT   MOVE     0    >DFF0____    (DFF)
```

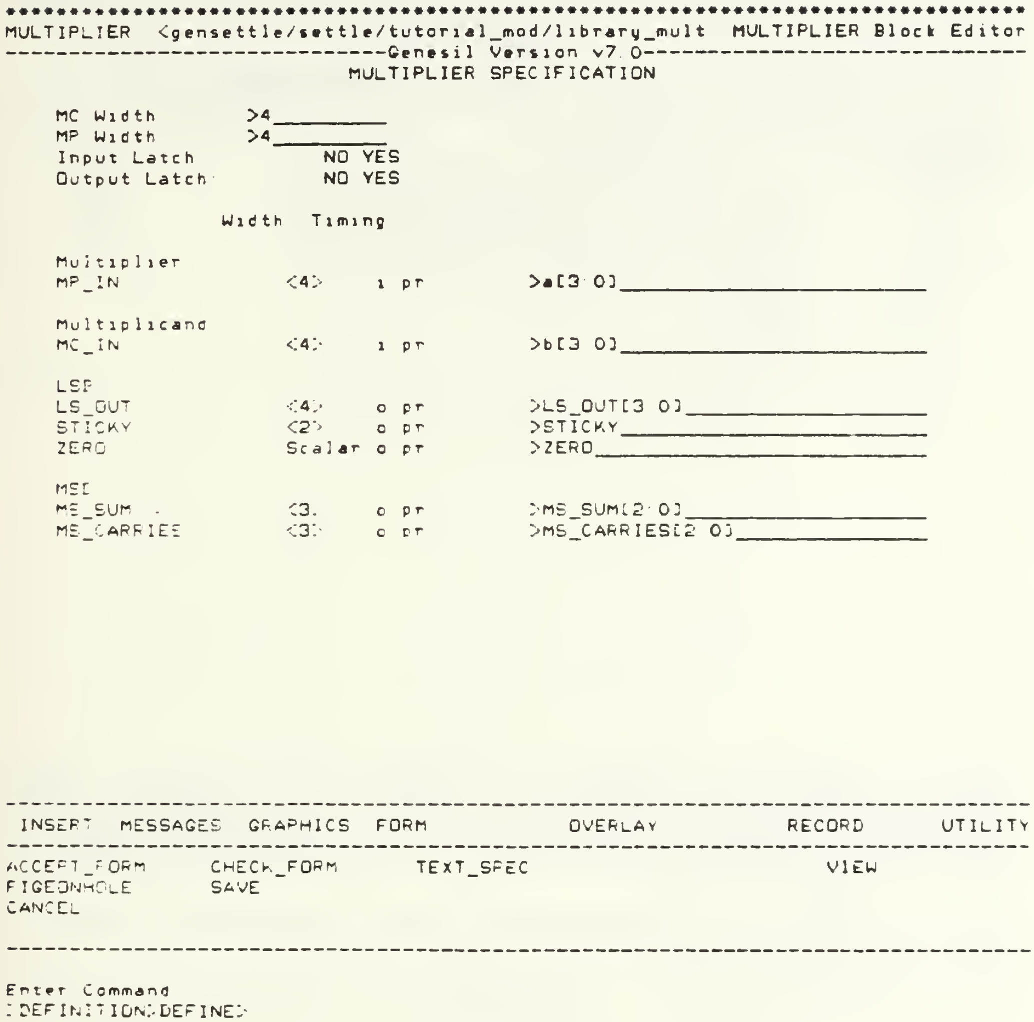
```
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_FORM       NEW              DELETE_GROUP     VIEW
PIGEONHOLE       SAVE             SIGNALS          EDIT_GROUP       TEXT_SPEC
CANCEL           TECH_CHECK       UNUSED

--------------------------------------------------------------------------------

:DEFINITION:DEFINE:
```

Figure A.39



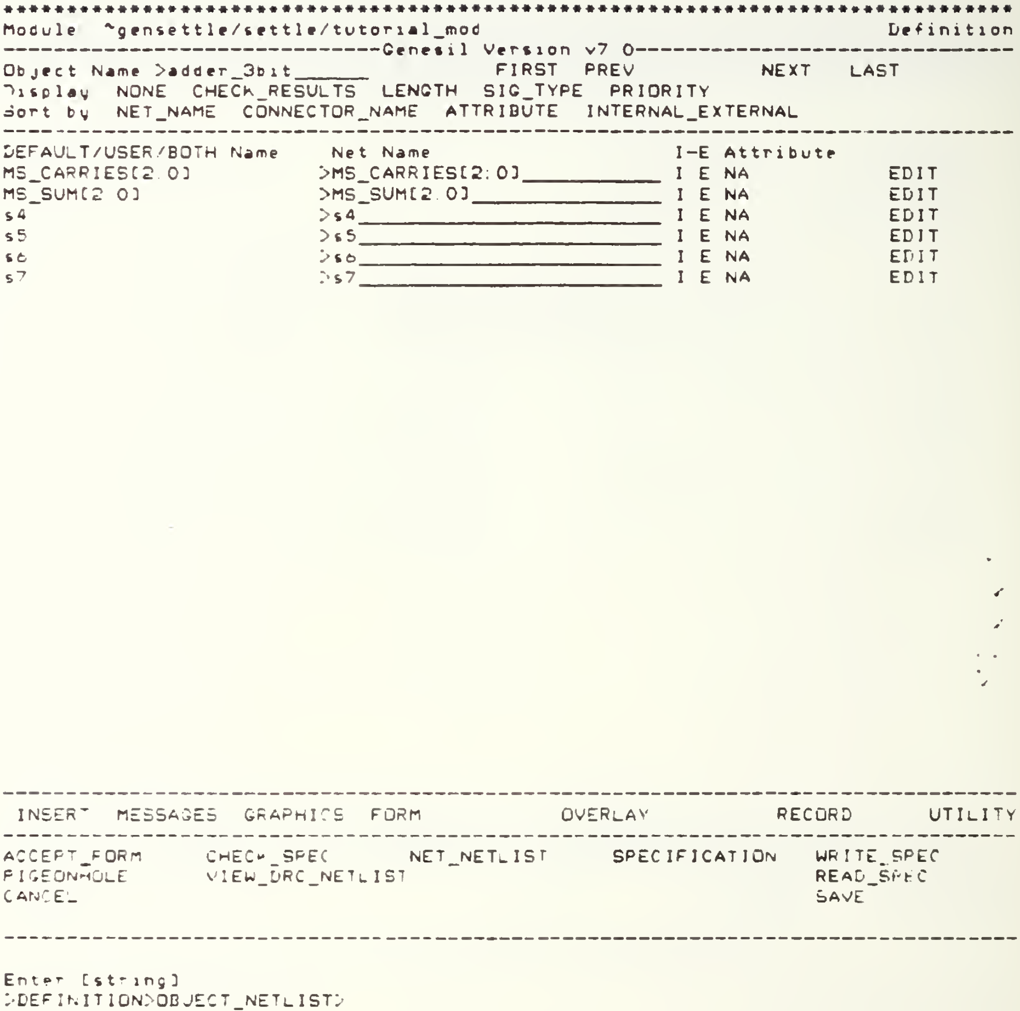
```
******************************************************************************
Random_Logic  <gensettle/settle/tutorial_mod/dff_8bit  Random Logic Block Editor
-------------------------------Genesil Version v7 0-------------------------------
                          Random Logic Block Specification

Block type    DFF
Block index   0
Name          >DFF0____
Width         >_8

                 Regime
Connector   Width  ! Timing
PHX           1      1 Phase X      >phase_a__________________________
PHY           1      1 Phase Y      >phase_b__________________________
IN            8      1 Vy(t-1)      >s7,s6,s5,s4,LS_OUT[3 0]__________
OUT           8      1 Sy(t)        >P7,P6,P5,P4,P3,P2,P1,P0__________
LOAD          1      1 Vy(t-1)      >TRUE_____________________________







------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
BACK                            CONVERT           EXPAND



------------------------------------------------------------------------------

>DEFINITION>DEFINE:
```

Figure A.40



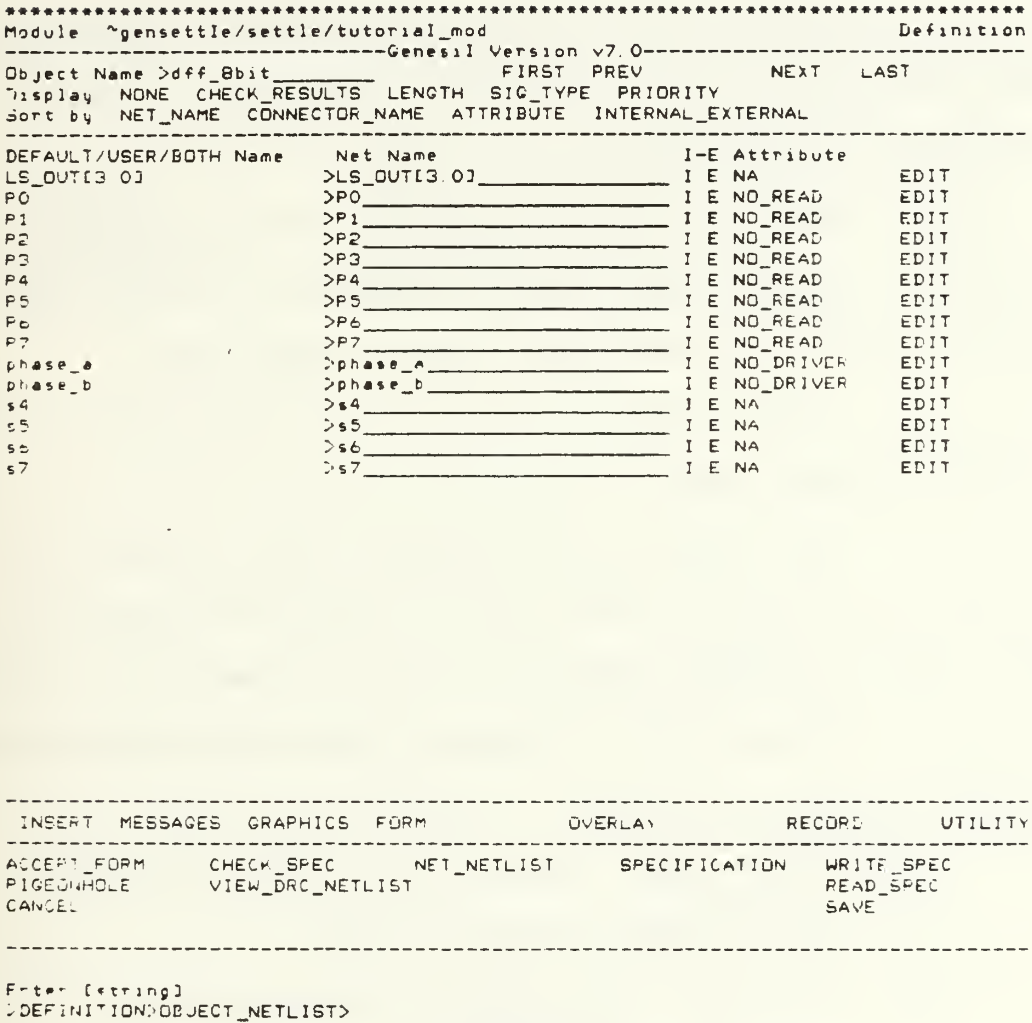
```
*******************************************************************************
MULTIPLIER  <gensettle/settle/tutorial_mod/library_mult  MULTIPLIER Block Editor
-----------------------------Genesil Version v7.0-------------------------------
                         MULTIPLIER SPECIFICATION

   MC Width        >4________
   MP Width        >4________
   Input Latch           NO YES
   Output Latch          NO YES

               Width  Timing

   Multiplier
   MP_IN              <4>     i pr        >a[3 0]______________________

   Multiplicand
   MC_IN              <4>     i pr        >b[3 0]______________________

   LSP
   LS_OUT             <4>     o pr        >LS_OUT[3 0]_________________
   STICKY             <2>     o pr        >STICKY______________________
   ZERO               Scalar  o pr        >ZERO________________________

   MSP
   MS_SUM             <3>     o pr        >MS_SUM[2 0]_________________
   MS_CARRIES         <3>     o pr        >MS_CARRIES[2 0]_____________




--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM         OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_FORM     TEXT_SPEC                       VIEW
PIGEONHOLE      SAVE
CANCEL

--------------------------------------------------------------------------------

Enter Command
<DEFINITION>DEFINE>
```

Figure A.41



```
**********************************************************************************
Module  ~gensettle/settle/tutorial_mod                                  Definition
------------------------------Genesil Version v7.0------------------------------
Object Name >adder_3bit______         FIRST  PREV          NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
--------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                      I-E Attribute
MS_CARRIES[2:0]           >MS_CARRIES[2:0]___________   I E NA           EDIT
MS_SUM[2:0]               >MS_SUM[2:0]_______________   I E NA           EDIT
s4                        >s4________________________   I E NA           EDIT
s5                        >s5________________________   I E NA           EDIT
s6                        >s6________________________   I E NA           EDIT
s7                        >s7________________________   I E NA           EDIT







--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC      NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                                READ_SPEC
CANCEL                                                          SAVE

--------------------------------------------------------------------------------

Enter [string]
>DEFINITION>OBJECT_NETLIST>
```

Figure A.42



```
*****************************************************************************
Module  ~gensettle/settle/tutorial_mod                              Definition
----------------------------Genesil Version v7.0-------------------------------
Object Name >dff_8bit_________          FIRST  PREV          NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                    I-E Attribute
LS_OUT[3 0]               >LS_OUT[3.0]_____________   I E NA            EDIT
P0                        >P0______________________   I E NO_READ       EDIT
P1                        >P1______________________   I E NO_READ       EDIT
P2                        >P2______________________   I E NO_READ       EDIT
P3                        >P3______________________   I E NO_READ       EDIT
P4                        >P4______________________   I E NO_READ       EDIT
P5                        >P5______________________   I E NO_READ       EDIT
P6                        >P6______________________   I E NO_READ       EDIT
P7                        >P7______________________   I E NO_READ       EDIT
phase_a                   >phase_a_________________   I E NO_DRIVER     EDIT
phase_b                   >phase_b_________________   I E NO_DRIVER     EDIT
s4                        >s4______________________   I E NA            EDIT
s5                        >s5______________________   I E NA            EDIT
s6                        >s6______________________   I E NA            EDIT
s7                        >s7______________________   I E NA            EDIT




------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC      NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                                READ_SPEC
CANCEL                                                          SAVE

------------------------------------------------------------------------------

Enter [string]
>DEFINITION>OBJECT_NETLIST>
```

Figure A.43

e. Select Object Name, and type in library_mult (Figure A.44). Do the same as in c above.

f. Now select SPECIFICATION (Figure A.44) to check the netlist. The system will state if valid or advise of errors which are listed by using VIEW_DRC_NETLIST.

3. The module must now be floorplanned. Simulation and timing analysis cannot be performed on a module prior to netlisting and floorplanning.

a. After netlist validation, the screen should show a module Definition menu. If not, go BACK or ACCEPT_FORM to return to the Definition menu. Select FLOOR_PLAN.

b. Figure A.45 should now be on the screen. Select PLACEMENT (Figure A.45).

c. Next select each unplaced block (Figure A.46) until all are placed. Blocks may be moved by hooking the object with the MOUSE. The rest of the commands are explained in detail in Chapter 6 of the System Description Applications Commands manual.

d. Next go BACK (Figure A.46) to Figure A.45.

e. Select PINOUT (Figure A.45).

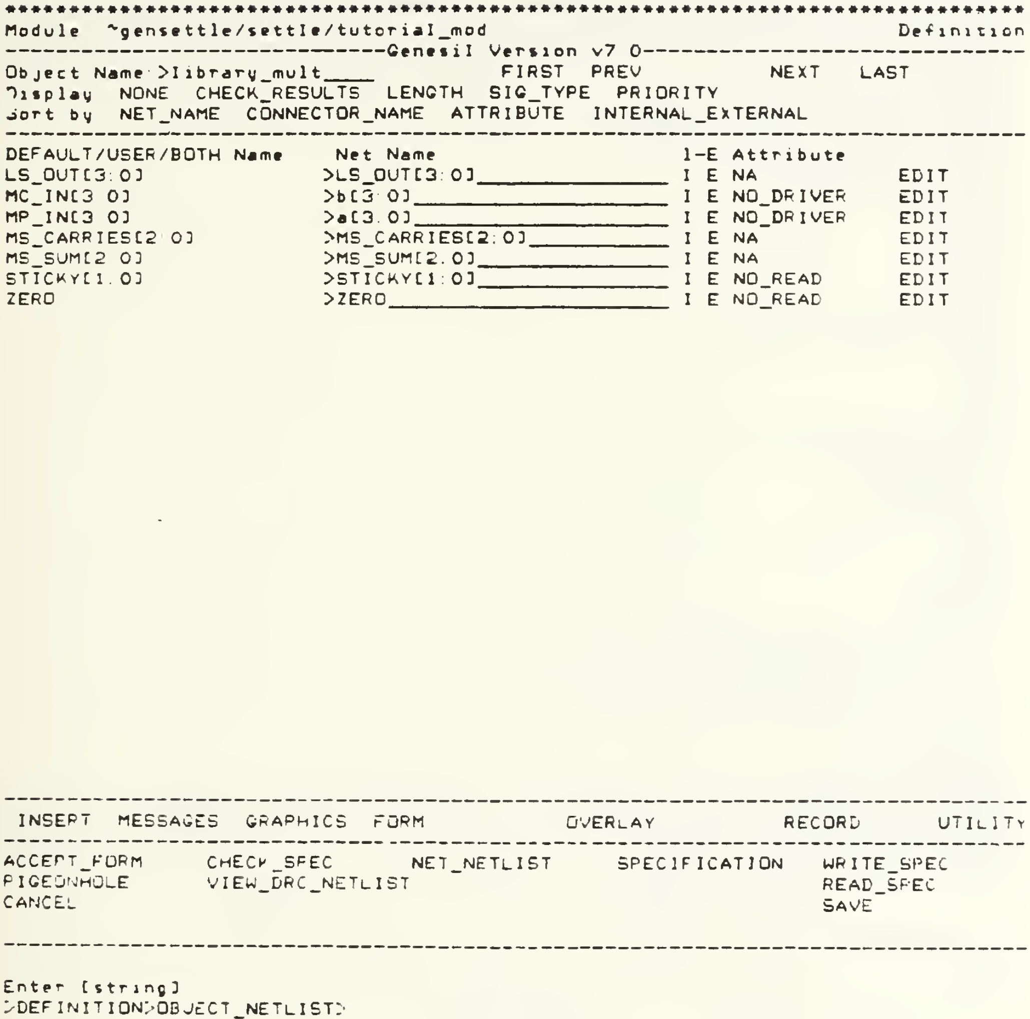
f. Now select AUTO_PINOUT (Figure A.47).

g. Go BACK (Figure A.47) to Figure A.45.

h. Select FUSION (Figure A.45).

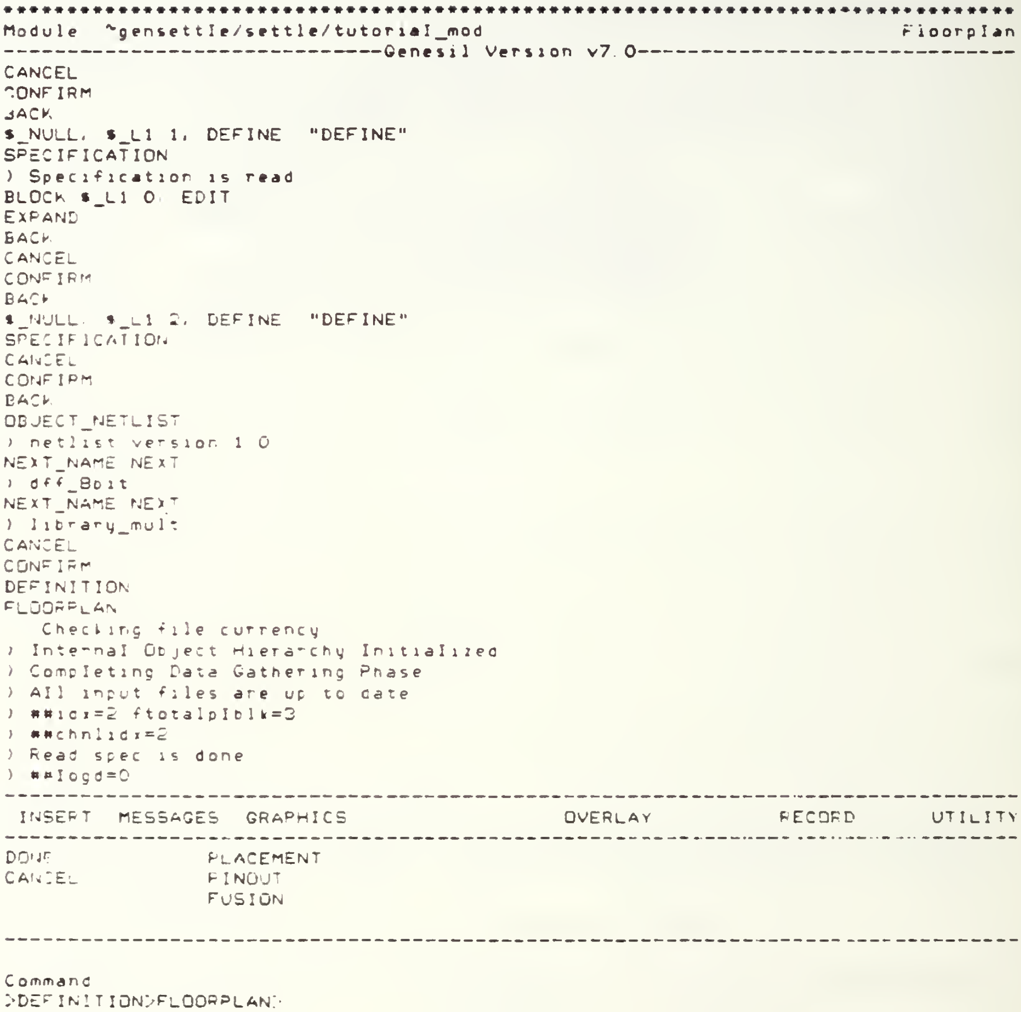
i. Select AUTO_FUSE (Figure A.48).

j. Now go BACK (Figure A.48) to A.45.



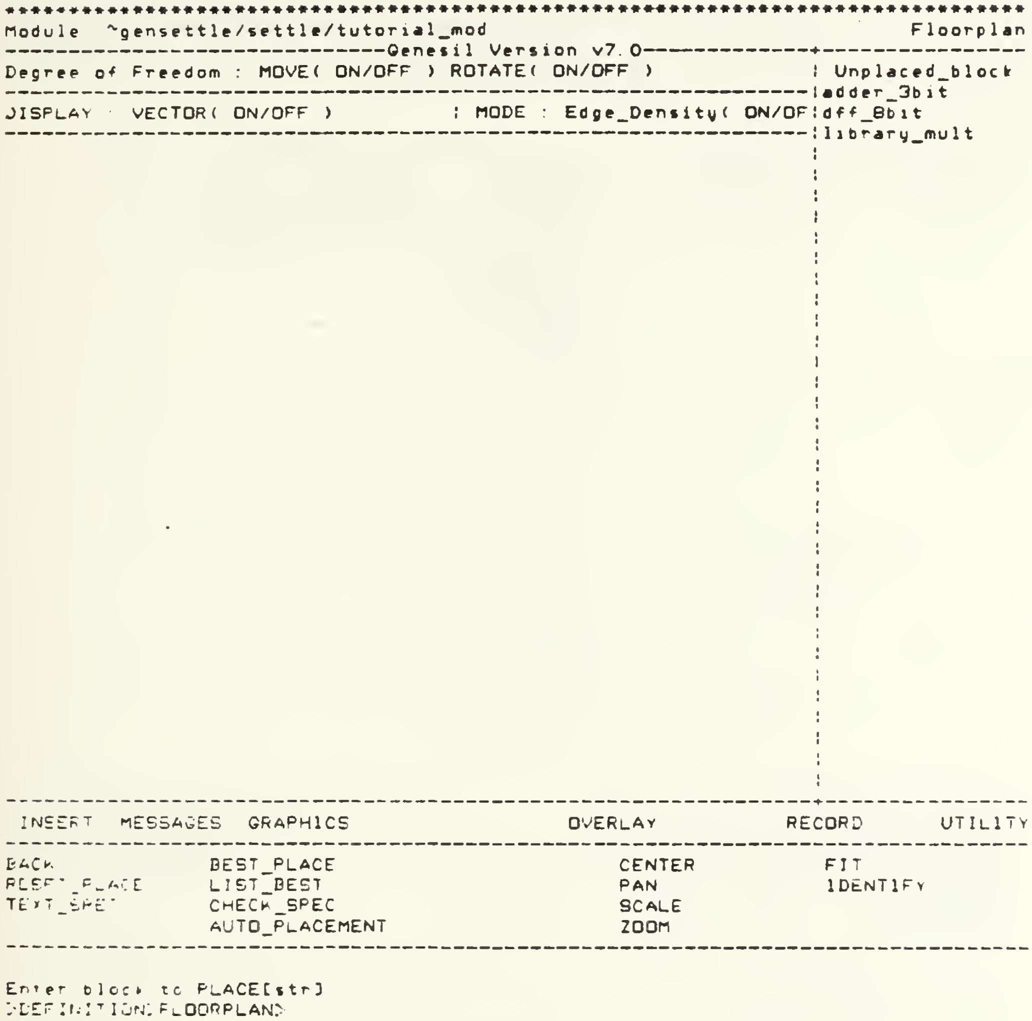
```
********************************************************************************
Module  ~gensettle/settle/tutorial_mod                                Definition
-----------------------------Genesil Version v7 0-------------------------------
Object Name >library_mult____          FIRST  PREV          NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
--------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                     1-E Attribute
LS_OUT[3:0]              >LS_OUT[3:0]______________  I E NA          EDIT
MC_IN[3 0]               >b[3 0]___________________  I E NO_DRIVER   EDIT
MP_IN[3 0]               >a[3.0]___________________  I E NO_DRIVER   EDIT
MS_CARRIES[2 0]          >MS_CARRIES[2:0]__________  I E NA          EDIT
MS_SUM[2 0]              >MS_SUM[2.0]______________  I E NA          EDIT
STICKY[1.0]              >STICKY[1:0]______________  I E NO_READ     EDIT
ZERO                     >ZERO_____________________  I E NO_READ     EDIT










--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY        RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC     NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                               READ_SPEC
CANCEL                                                         SAVE

--------------------------------------------------------------------------------

Enter [string]
>DEFINITION>OBJECT_NETLIST>
```

Figure A.44



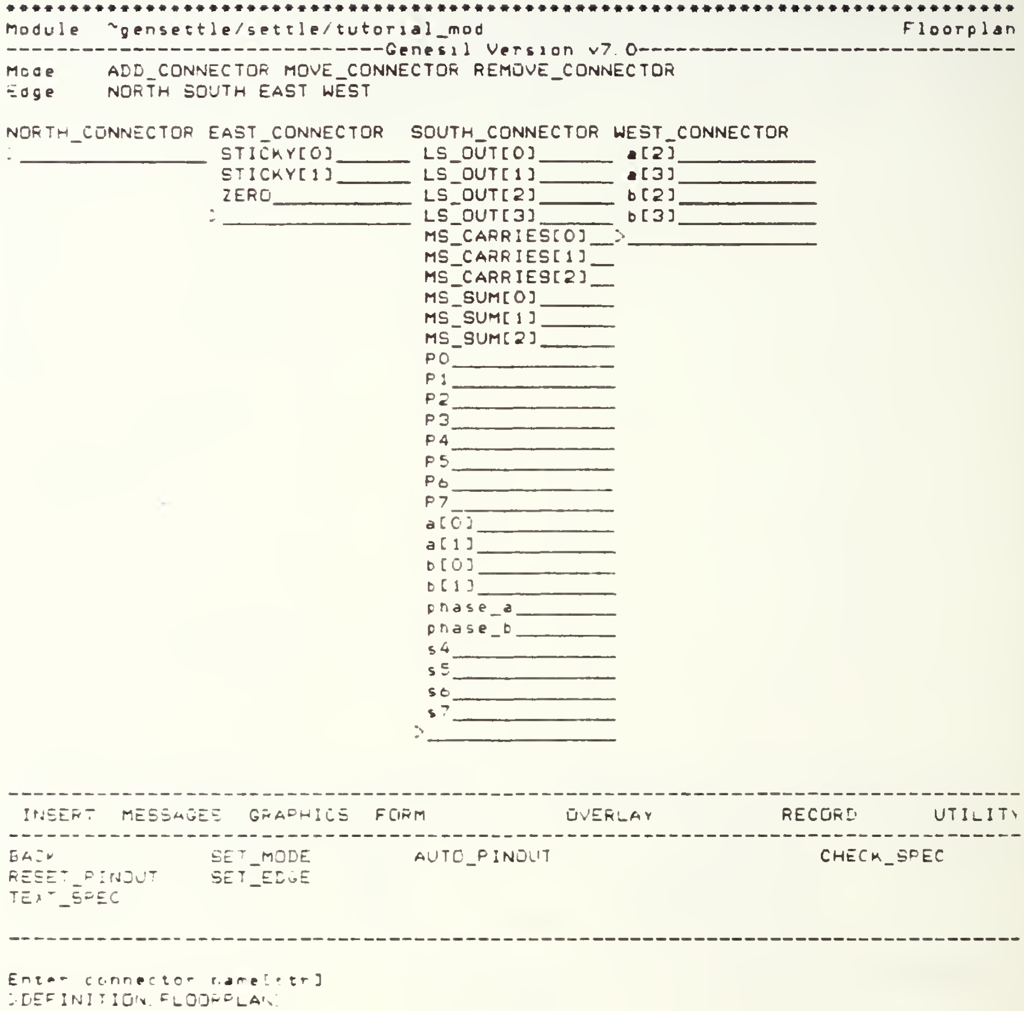
```
****************************************************************************
Module  ~gensettle/settle/tutorial_mod                              Floorplan
-----------------------------Genesil Version v7.0----------------------------
CANCEL
CONFIRM
BACK
$_NULL, $_L1 1, DEFINE  "DEFINE"
SPECIFICATION
) Specification is read
BLOCK $_L1 0, EDIT
EXPAND
BACK
CANCEL
CONFIRM
BACK
$_NULL, $_L1 2, DEFINE  "DEFINE"
SPECIFICATION
CANCEL
CONFIRM
BACK
OBJECT_NETLIST
) netlist version 1.0
NEXT_NAME NEXT
) dff_8bit
NEXT_NAME NEXT
) library_mult
CANCEL
CONFIRM
DEFINITION
FLOORPLAN
   Checking file currency
) Internal Object Hierarchy Initialized
) Completing Data Gathering Phase
) All input files are up to date
) ##idx=2 ftotalplblk=3
) ##chnlidx=2
) Read spec is done
) ##logd=0
-----------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                  OVERLAY          RECORD     UTILITY
-----------------------------------------------------------------------------
DONE             PLACEMENT
CANCEL           PINOUT
                 FUSION

-----------------------------------------------------------------------------

Command
>DEFINITION>FLOORPLAN>
```

Figure A.45



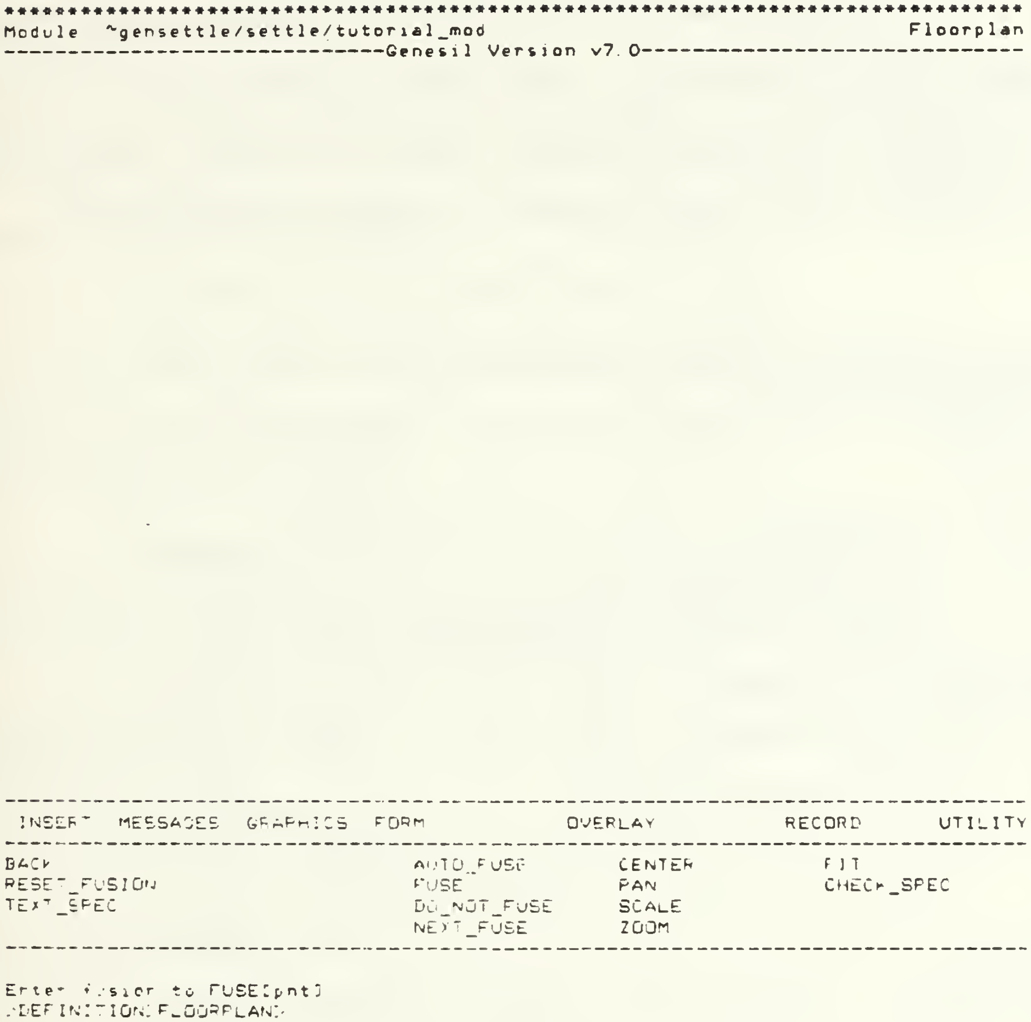
```
******************************************************************************
Module  ~gensettle/settle/tutorial_mod                                Floorplan
-----------------------------Genesil Version v7.0-------------+----------------
Degree of Freedom : MOVE( ON/OFF ) ROTATE( ON/OFF )            | Unplaced_block
---------------------------------------------------------------|adder_3bit
DISPLAY : VECTOR( ON/OFF )          : MODE : Edge_Density( ON/OF|dff_8bit
---------------------------------------------------------------|library_mult
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
                                                               |
---------------------------------------------------------------+---------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY         RECORD      UTILITY
-------------------------------------------------------------------------------
BACK            BEST_PLACE                     CENTER          FIT
RESET_PLACE     LIST_BEST                      PAN             IDENTIFY
TEXT_SPEC       CHECK_SPEC                     SCALE
                AUTO_PLACEMENT                 ZOOM
-------------------------------------------------------------------------------

Enter block to PLACE[str]
DEFINITION:FLOORPLAN>
```

Figure A.46



```
*************************************************************************
Module  ~gensettle/settle/tutorial_mod                         Floorplan
----------------------------Genesil Version v7.0-------------------------
Mode    ADD_CONNECTOR MOVE_CONNECTOR REMOVE_CONNECTOR
Edge    NORTH SOUTH EAST WEST

NORTH_CONNECTOR EAST_CONNECTOR  SOUTH_CONNECTOR WEST_CONNECTOR
: _____________ STICKY[0]______ LS_OUT[0]______ a[2]__________
                STICKY[1]______ LS_OUT[1]______ a[3]__________
                ZERO___________ LS_OUT[2]______ b[2]__________
              : _______________ LS_OUT[3]______ b[3]__________
                                MS_CARRIES[0]__>______________
                                MS_CARRIES[1]__
                                MS_CARRIES[2]__
                                MS_SUM[0]______
                                MS_SUM[1]______
                                MS_SUM[2]______
                                P0_____________
                                P1_____________
                                P2_____________
                                P3_____________
                                P4_____________
                                P5_____________
                                P6_____________
                                P7_____________
                                a[0]___________
                                a[1]___________
                                b[0]___________
                                b[1]___________
                                phase_a________
                                phase_b________
                                s4_____________
                                s5_____________
                                s6_____________
                                s7_____________
                               >_______________


-------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD    UTILITY
-------------------------------------------------------------------------
BACK            SET_MODE        AUTO_PINOUT                    CHECK_SPEC
RESET_PINOUT    SET_EDGE
TEXT_SPEC

-------------------------------------------------------------------------

Enter connector name[str]
>DEFINITION.FLOORPLAN
```

Figure A.47



```
*******************************************************************************
Module  ~gensettle/settle/tutorial_mod                                Floorplan
------------------------------Genesil Version v7.0------------------------------

















--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK                                     AUTO_FUSE      CENTER          FIT
RESET_FUSION                             FUSE           PAN             CHECK_SPEC
TEXT_SPEC                                DO_NOT_FUSE    SCALE
                                         NEXT_FUSE      ZOOM
--------------------------------------------------------------------------------

Enter fusion to FUSE[pnt]
<DEFINITION:FLOORPLAN>
```

Figure A.48

k. Select DONE (Figure A.45), followed by CONFIRM. If all goes well, the floorplan will be complete.

4. Now simulation and timing analysis can be performed as described in the previous sections.

E. TUTOR_CHIP CHIP

This section is a tutorial for a top-down chip design of a chip named tutor_chip. The chip consists of tutorblk_1 and tutorblk-2, a clock, input/out pads, a VSS pad, and VDD pad. A schematic with signal names is shown in Figure A.49.

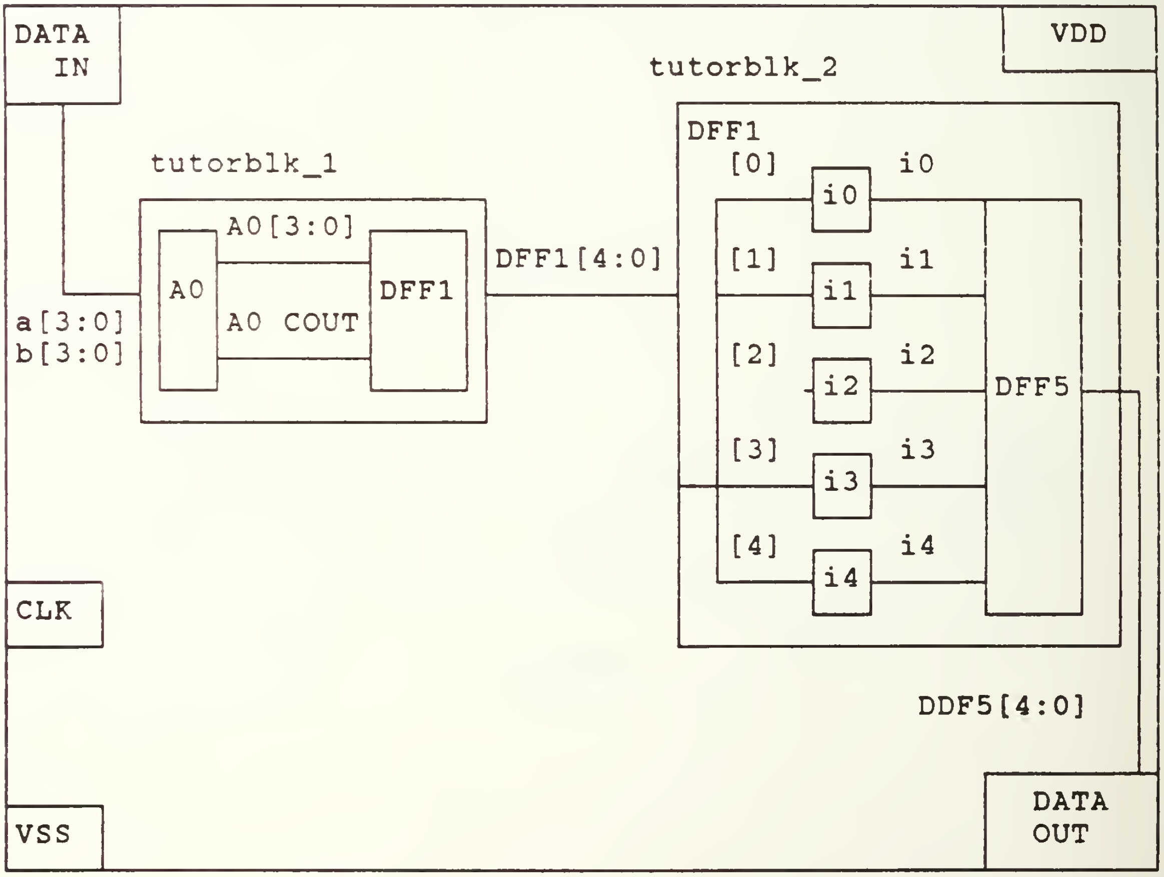


Figure A.49 Tutor_Chip Chip

1. Definition and Specification:

a. Proceed as in the previous sections down through ATTACH NEW. Now select CHIP from the Executive menu (Figure A.50).

b. Name the chip tutor_chip when prompted, and attach it to the tree.

c. Select DEFINITION and HEADER. On the Chip Header form (Figure A.51) select Fab line VTC_CP10A and leave the Package Type as NO_PACKAGE.

d. Select ACCEPT_FORM (Figure A.51).

e. Now select SPECIFICATION from the Definition menu and the screen should show a blank Chip Specification form (Figure A.52).

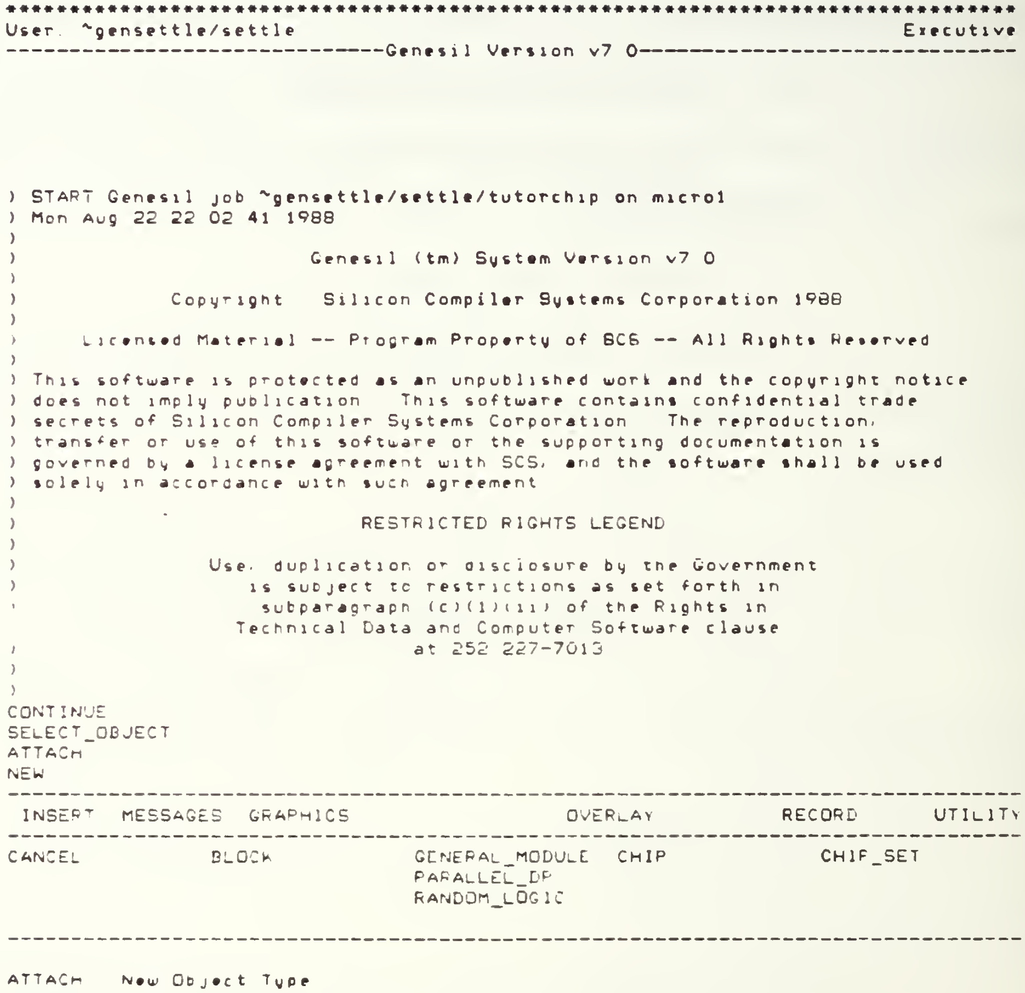
f. Next select ATTACH_NEW (Figure A.52) and BLOCK (Figure A.53). Name the object data_in when prompted.

g. Just like the module, each sub-object on the chip must be defined by using the appropriate Header and Specification forms.

h. Select DEFINE next to data_in on the Chip Specification menu.

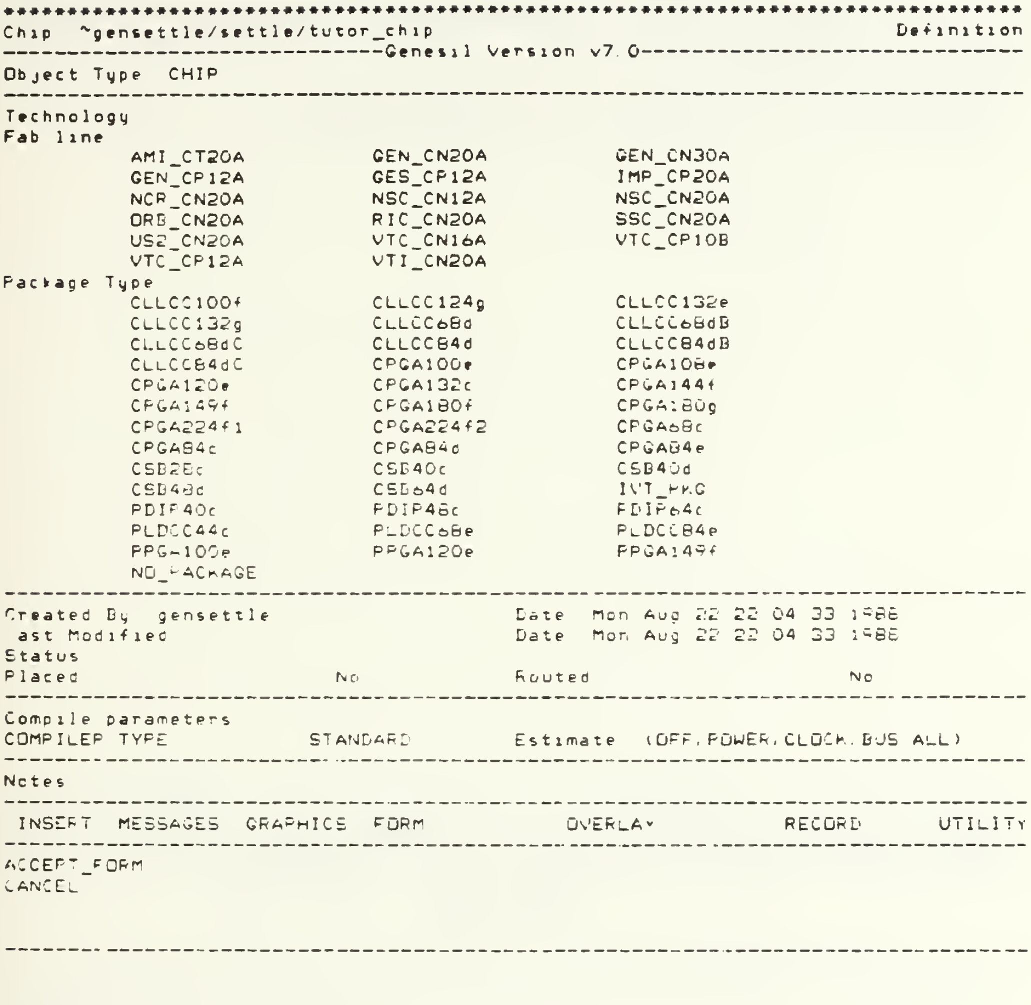
i. Select HEADER from the next screen. Select PAD from the Header form for Function type (Figure A.54) and CONFIRM. Select ACCEPT_FORM.

j. Next select SPECIFICATION and the screen should show a PAD Functional Specification menu (Figure A.55).



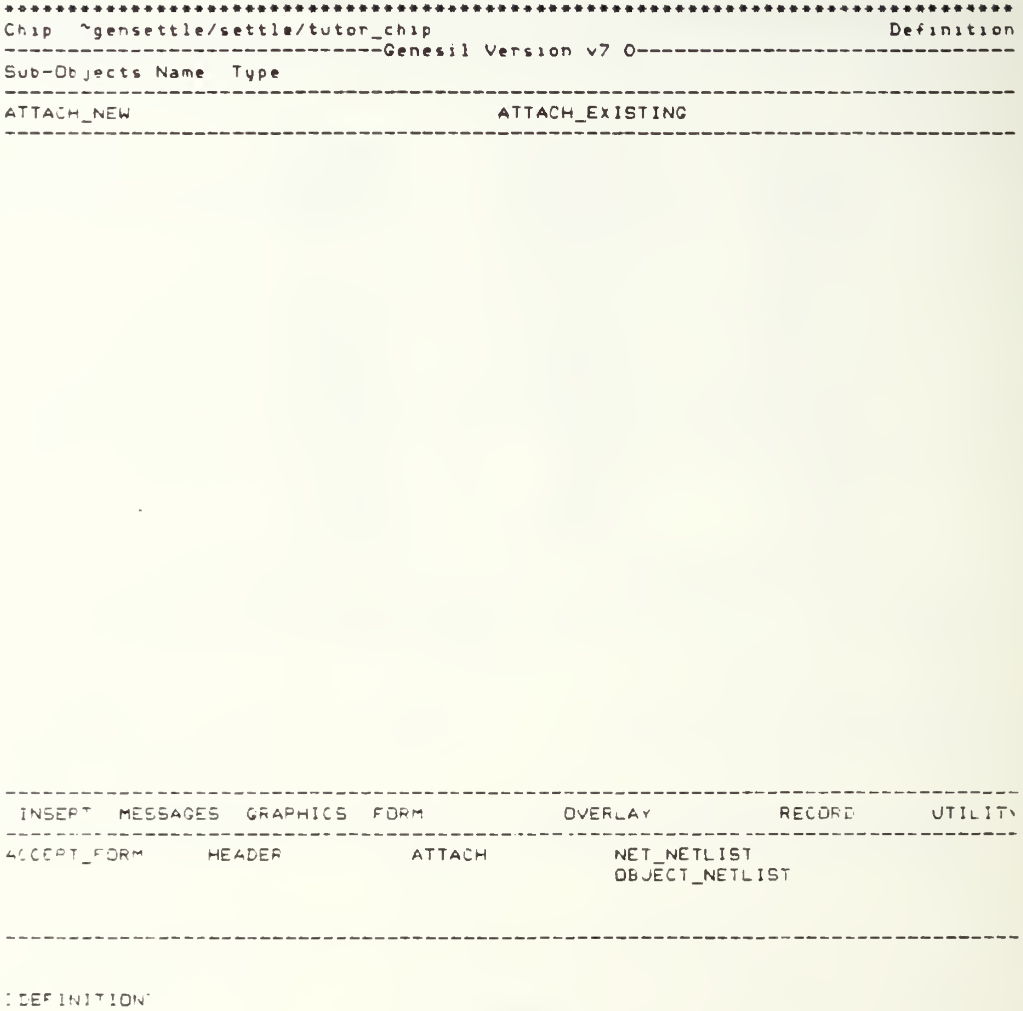
```
********************************************************************************
User. ~gensettle/settle                                                Executive
------------------------------Genesil Version v7 0------------------------------



) START Genesil job ~gensettle/settle/tutorchip on micro1
) Mon Aug 22 22 02 41 1988
)
)                        Genesil (tm) System Version v7 0
)
)             Copyright   Silicon Compiler Systems Corporation 1988
)
)     Licensed Material -- Program Property of SCS -- All Rights Reserved
)
) This software is protected as an unpublished work and the copyright notice
) does not imply publication   This software contains confidential trade
) secrets of Silicon Compiler Systems Corporation   The reproduction,
) transfer or use of this software or the supporting documentation is
) governed by a license agreement with SCS, and the software shall be used
) solely in accordance with such agreement
)
)                            RESTRICTED RIGHTS LEGEND
)
)                 Use, duplication or disclosure by the Government
)                    is subject to restrictions as set forth in
)                     subparagraph (c)(1)(ii) of the Rights in
)                   Technical Data and Computer Software clause
)                                at 252 227-7013
)
)
CONTINUE
SELECT_OBJECT
ATTACH
NEW
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                   OVERLAY            RECORD    UTILITY
--------------------------------------------------------------------------------
CANCEL           BLOCK            GENERAL_MODULE  CHIP             CHIP_SET
                                  PARALLEL_DP
                                  RANDOM_LOGIC

--------------------------------------------------------------------------------

ATTACH   New Object Type
>SELECT_OBJECT>ATTACH>NEW>
```

Figure A.50



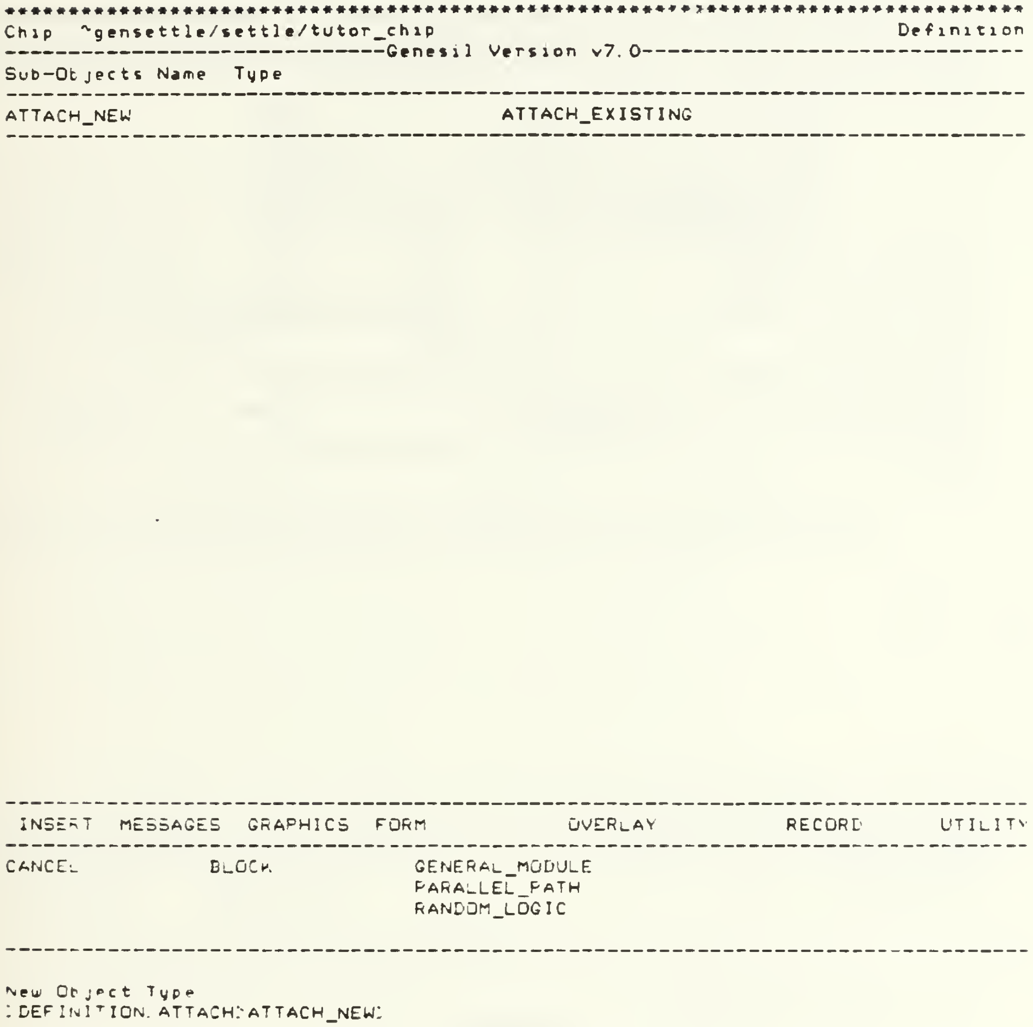
```
******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                  Definition
------------------------------Genesil Version v7.0----------------------------
Object Type  CHIP
------------------------------------------------------------------------------
Technology
Fab line
         AMI_CT20A           GEN_CN20A           GEN_CN30A
         GEN_CP12A           GES_CP12A           IMP_CP20A
         NCR_CN20A           NSC_CN12A           NSC_CN20A
         ORB_CN20A           RIC_CN20A           SSC_CN20A
         US2_CN20A           VTC_CN16A           VTC_CP10B
         VTC_CP12A           VTI_CN20A
Package Type
         CLLCC100f           CLLCC124g           CLLCC132e
         CLLCC132g           CLLCC68d            CLLCC68dB
         CLLCC68dC           CLLCC84d            CLLCC84dB
         CLLCC84dC           CPGA100e            CPGA108e
         CPGA120e            CPGA132c            CPGA144f
         CPGA149f            CPGA180f            CPGA180g
         CPGA224f1           CPGA224f2           CPGA68c
         CPGA84c             CPGA84d             CPGA84e
         CSB28c              CSB40c              CSB40d
         CSB48d              CSB64d              INT_PKG
         PDIP40c             PDIP48c             PDIP64c
         PLDCC44c            PLDCC68e            PLDCC84e
         PPGA100e            PPGA120e            PPGA149f
         NO_PACKAGE
------------------------------------------------------------------------------
Created By  gensettle                       Date  Mon Aug 22 22 04 33 1988
 ast Modified                               Date  Mon Aug 22 22 04 33 1988
Status
Placed                         No           Routed                   No
------------------------------------------------------------------------------
Compile parameters
COMPILER TYPE           STANDARD            Estimate  (OFF,POWER,CLOCK,BUS ALL)
------------------------------------------------------------------------------
Notes
------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD     UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM
CANCEL


------------------------------------------------------------------------------


>DEFINITION>HEADER>
```

Figure A.51



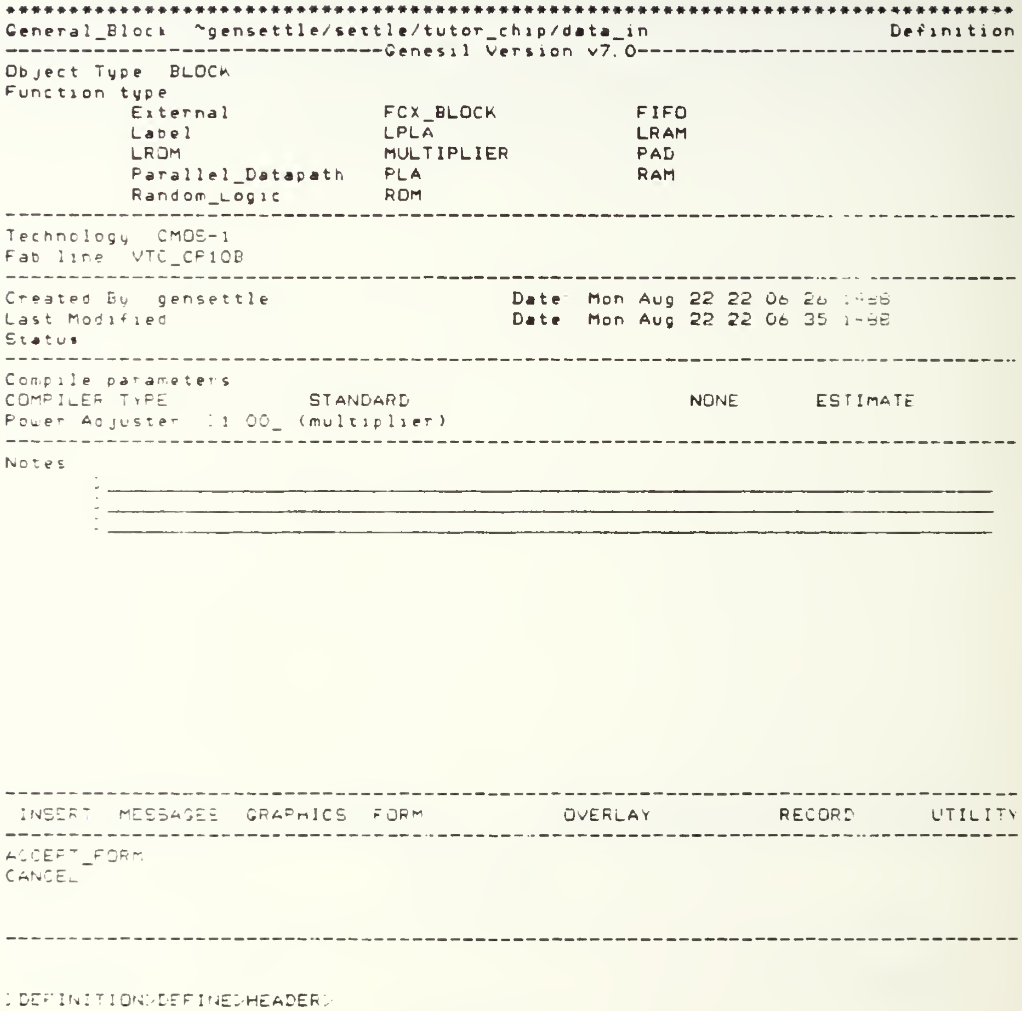
```
********************************************************************************
Chip  ~gensettle/settle/tutor_chip                                   Definition
------------------------------Genesil Version v7 0------------------------------
Sub-Objects Name  Type
--------------------------------------------------------------------------------
ATTACH_NEW                                     ATTACH_EXISTING
--------------------------------------------------------------------------------




--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY            RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      HEADER          ATTACH           NET_NETLIST
                                                  OBJECT_NETLIST

--------------------------------------------------------------------------------

[DEFINITION]
```

Figure A.52



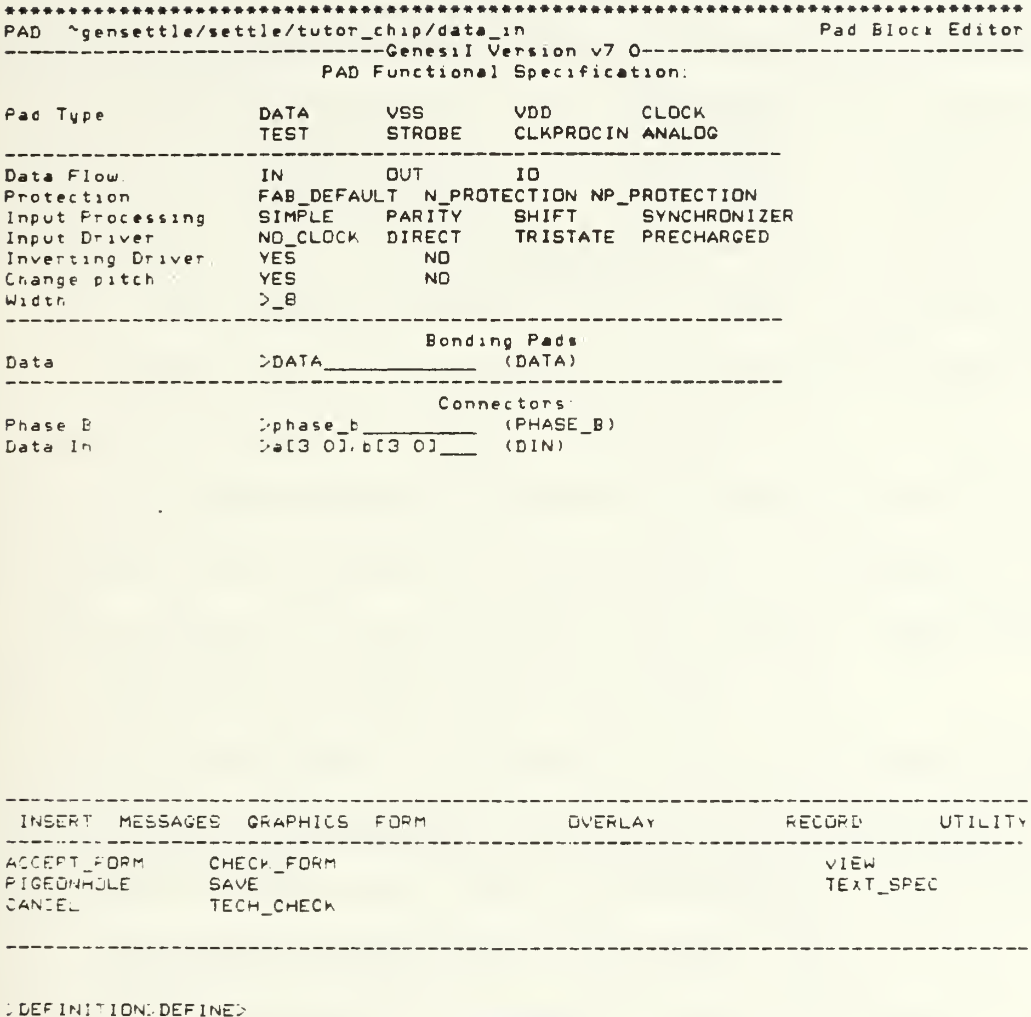
```
********************************************************************************
Chip  ~gensettle/settle/tutor_chip                                    Definition
------------------------------Genesil Version v7.0------------------------------
Sub-Objects Name  Type
--------------------------------------------------------------------------------
ATTACH_NEW                                     ATTACH_EXISTING
--------------------------------------------------------------------------------










--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL          BLOCK           GENERAL_MODULE
                                PARALLEL_PATH
                                RANDOM_LOGIC

--------------------------------------------------------------------------------

New Object Type
[DEFINITION.ATTACH>ATTACH_NEW]
```

Figure A.53



```
*******************************************************************************
General_Block  ~gensettle/settle/tutor_chip/data_in                  Definition
------------------------------Genesil Version v7.0------------------------------
Object Type  BLOCK
Function type
          External             FCX_BLOCK            FIFO
          Label                LPLA                 LRAM
          LROM                 MULTIPLIER           PAD
          Parallel_Datapath    PLA                  RAM
          Random_Logic         ROM
--------------------------------------------------------------------------------
Technology  CMOS-1
Fab line  VTC_CP10B
--------------------------------------------------------------------------------
Created By  gensettle                     Date  Mon Aug 22 22 06 26 1988
Last Modified                             Date  Mon Aug 22 22 06 35 1988
Status
--------------------------------------------------------------------------------
Compile parameters
COMPILER TYPE              STANDARD                       NONE      ESTIMATE
Power Adjuster  1 00_ (multiplier)
--------------------------------------------------------------------------------
Notes
       : ______________________________________________________________________
       : ______________________________________________________________________
       : ______________________________________________________________________







--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD        UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM
CANCEL


--------------------------------------------------------------------------------

<DEFINITION><DEFINED><HEADER>
```

Figure A.54



```
********************************************************************************
PAD  ~gensettle/settle/tutor_chip/data_in                       Pad Block Editor
-----------------------------Genesil Version v7 0-------------------------------
                          PAD Functional Specification:

Pad Type              DATA      VSS       VDD       CLOCK
                      TEST      STROBE    CLKPROCIN ANALOG
--------------------------------------------------------------
Data Flow:            IN        OUT       IO
Protection            FAB_DEFAULT  N_PROTECTION NP_PROTECTION
Input Processing      SIMPLE    PARITY    SHIFT     SYNCHRONIZER
Input Driver          NO_CLOCK  DIRECT    TRISTATE  PRECHARGED
Inverting Driver      YES          NO
Change pitch          YES          NO
Width                 >_8
--------------------------------------------------------------
                                 Bonding Pads:
Data                  >DATA____________   (DATA)
--------------------------------------------------------------
                                  Connectors:
Phase B               >phase_b_________   (PHASE_B)
Data In               >a[3 0],b[3 0]___   (DIN)




--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_FORM                                          VIEW
PIGEONHOLE       SAVE                                                TEXT_SPEC
CANCEL           TECH_CHECK

--------------------------------------------------------------------------------

>DEFINITION:DEFINED>
```

Figure A.55

Select IN for Data Flow. All other specifications are defaults except width, Phase B, and Data In. Fill these to match Figure A.55.

k. Now select ACCEPT_FORM (Figure A.55). Go BACK to the Chip Specification form (Figure A.56). Figure A.56 is a complete Chip Specification form for tutor_chip. Your Chip Specification form should have only a data-in PAD on it at this time.

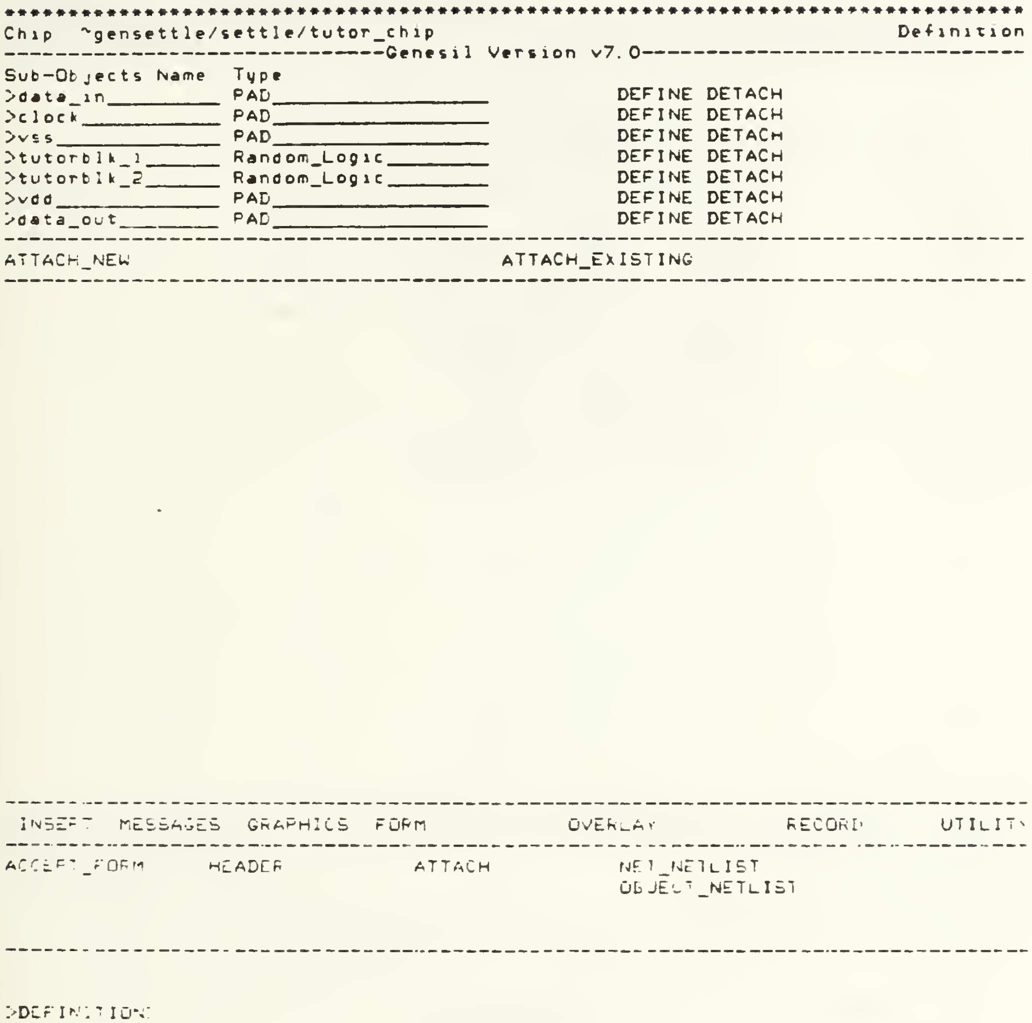
l. For each of the remaining PADS (clock, vss, vdd, and data_out), using steps f through j except for the following PAD Functional Specification form changes:

(1) Clock PAD: Select CLOCK for Pad type. All other specifications are defaults except PHASE_A and PHASE_B. type in phase_a, and phase_b in accordance with Figure A.57. ACCEPT_FORM (Figure A.57) and continue to the next PAD.

(2) VSS PAD: Select VSS for Pad Type. All other inputs are left to defaults. ACCEPT_FORM (Figure A.58) and continue to the next PAD. The Chip Specification form should now look like Figure A.59. The order is not important.

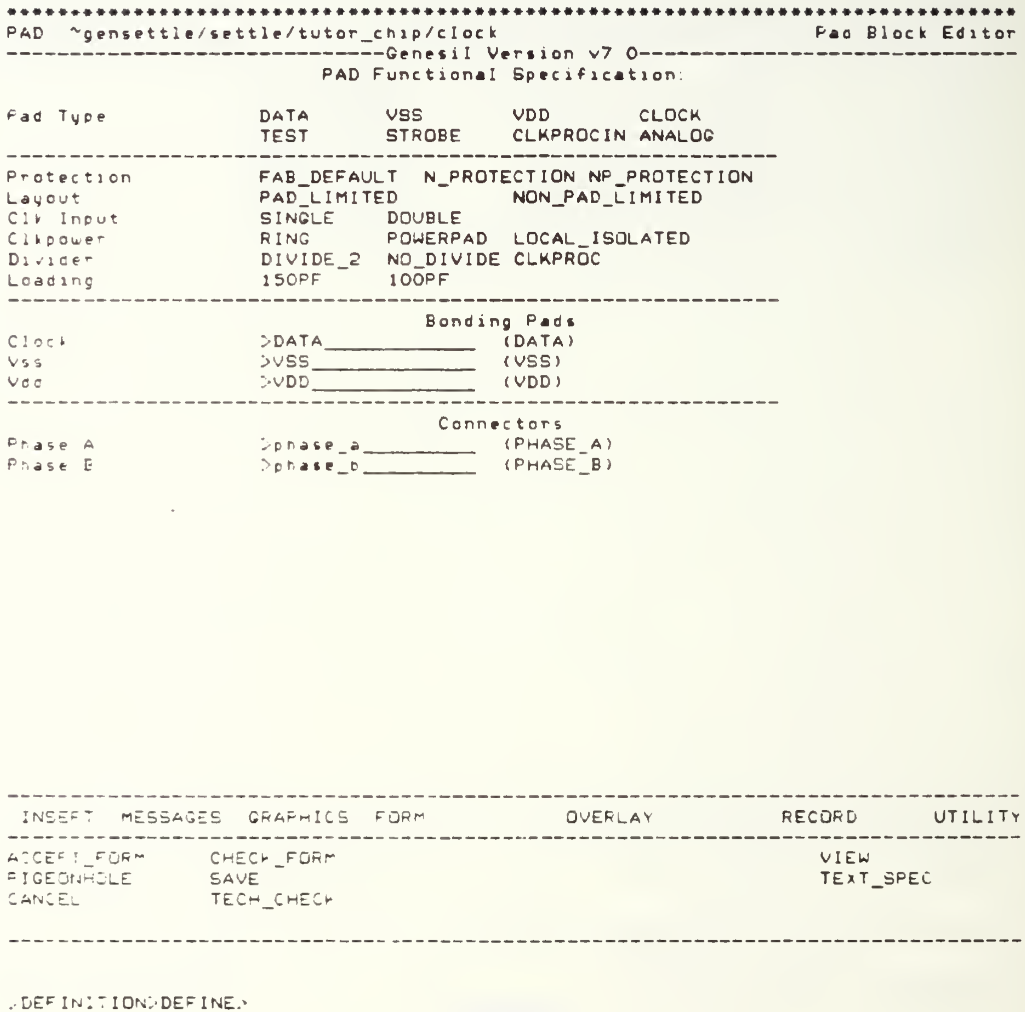
(3) Vdd PAD: Select vdd for PAD type. All other inputs are left to defaults. ACCEPT_FORM (Figure A.60) and continue to the next PAD.

(4) data_out PAD: Select OUT for Data Flow. All other specifications are defaults except width, Phase A, Phase B, and Data Out. Fill these in to match Figure A.61. ACCEPT_FORM and go BACK to the Chip Specification form.



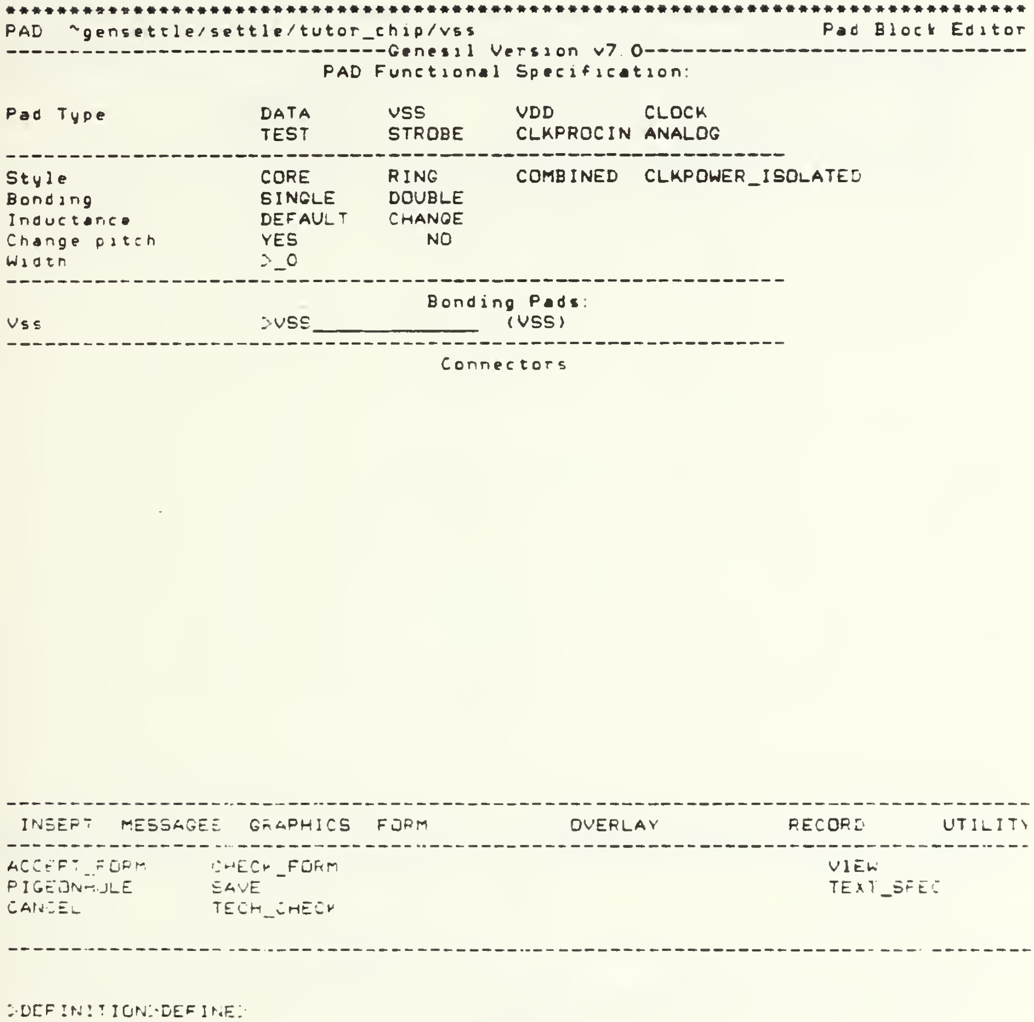
```
*****************************************************************************
Chip  ~gensettle/settle/tutor_chip                                  Definition
-----------------------------Genesil Version v7.0-----------------------------
Sub-Objects Name  Type
>data_in_________ PAD_________________        DEFINE DETACH
>clock___________ PAD_________________        DEFINE DETACH
>vss_____________ PAD_________________        DEFINE DETACH
>tutorblk_1______ Random_Logic________        DEFINE DETACH
>tutorblk_2______ Random_Logic________        DEFINE DETACH
>vdd_____________ PAD_________________        DEFINE DETACH
>data_out________ PAD_________________        DEFINE DETACH
------------------------------------------------------------------------------
ATTACH_NEW                          ATTACH_EXISTING
------------------------------------------------------------------------------




------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM     HEADER         ATTACH          NET_NETLIST
                                               OBJECT_NETLIST


------------------------------------------------------------------------------

>DEFINITION
```

Figure A.56



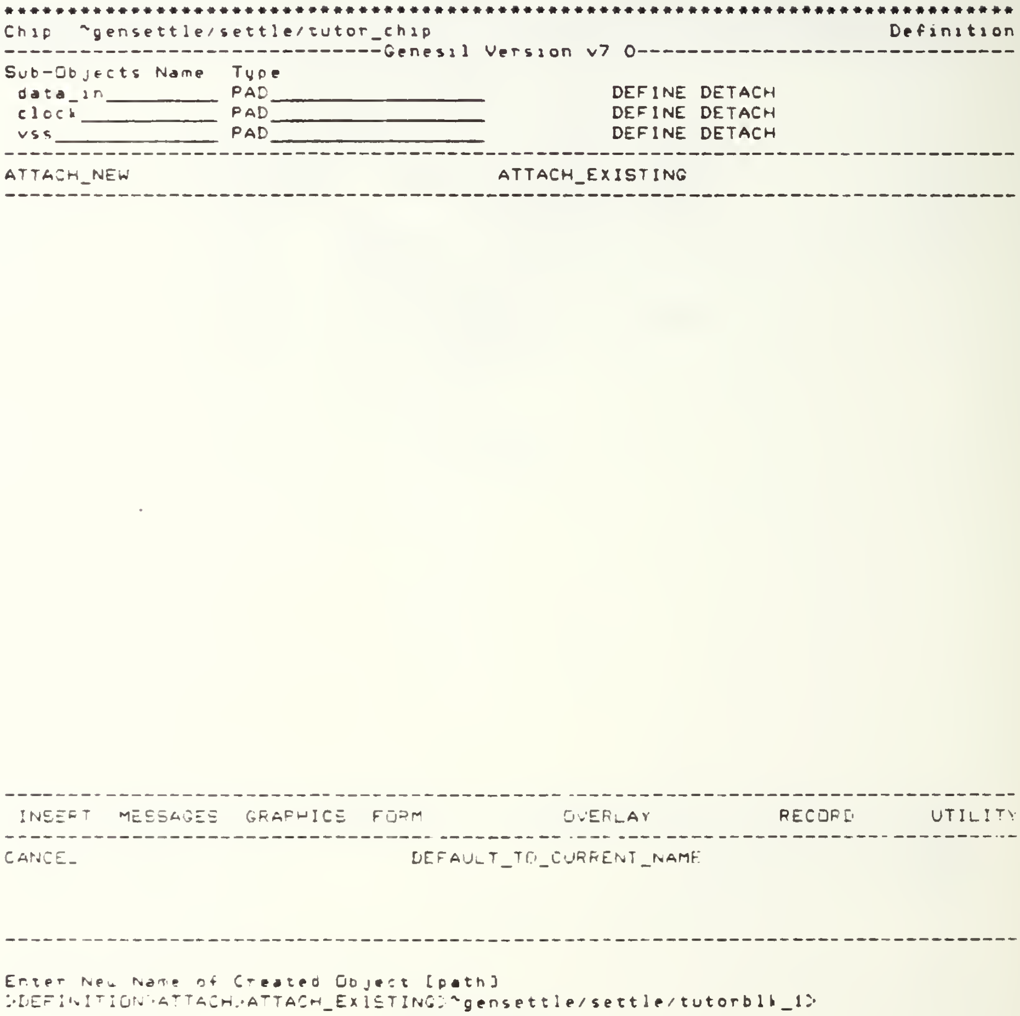
```
********************************************************************************
PAD  ~gensettle/settle/tutor_chip/clock                          Pad Block Editor
------------------------------Genesil Version v7 0------------------------------
                         PAD Functional Specification:

Pad Type                DATA      VSS       VDD       CLOCK
                        TEST      STROBE    CLKPROCIN ANALOG
-------------------------------------------------------------
Protection              FAB_DEFAULT  N_PROTECTION NP_PROTECTION
Layout                  PAD_LIMITED         NON_PAD_LIMITED
Clk Input               SINGLE    DOUBLE
Clkpower                RING      POWERPAD  LOCAL_ISOLATED
Divider                 DIVIDE_2  NO_DIVIDE CLKPROC
Loading                 150PF     100PF
-------------------------------------------------------------
                                    Bonding Pads
Clock                   >DATA__________     (DATA)
Vss                     >VSS___________     (VSS)
Vdd                     >VDD___________     (VDD)
-------------------------------------------------------------
                                     Connectors
Phase A                 >phase_a_______     (PHASE_A)
Phase B                 >phase_b_______     (PHASE_B)




--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD       UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_FORM                                      VIEW
PIGEONHOLE      SAVE                                            TEXT_SPEC
CANCEL          TECH_CHECK

--------------------------------------------------------------------------------

>DEFINITION>DEFINE>
```

Figure A.57



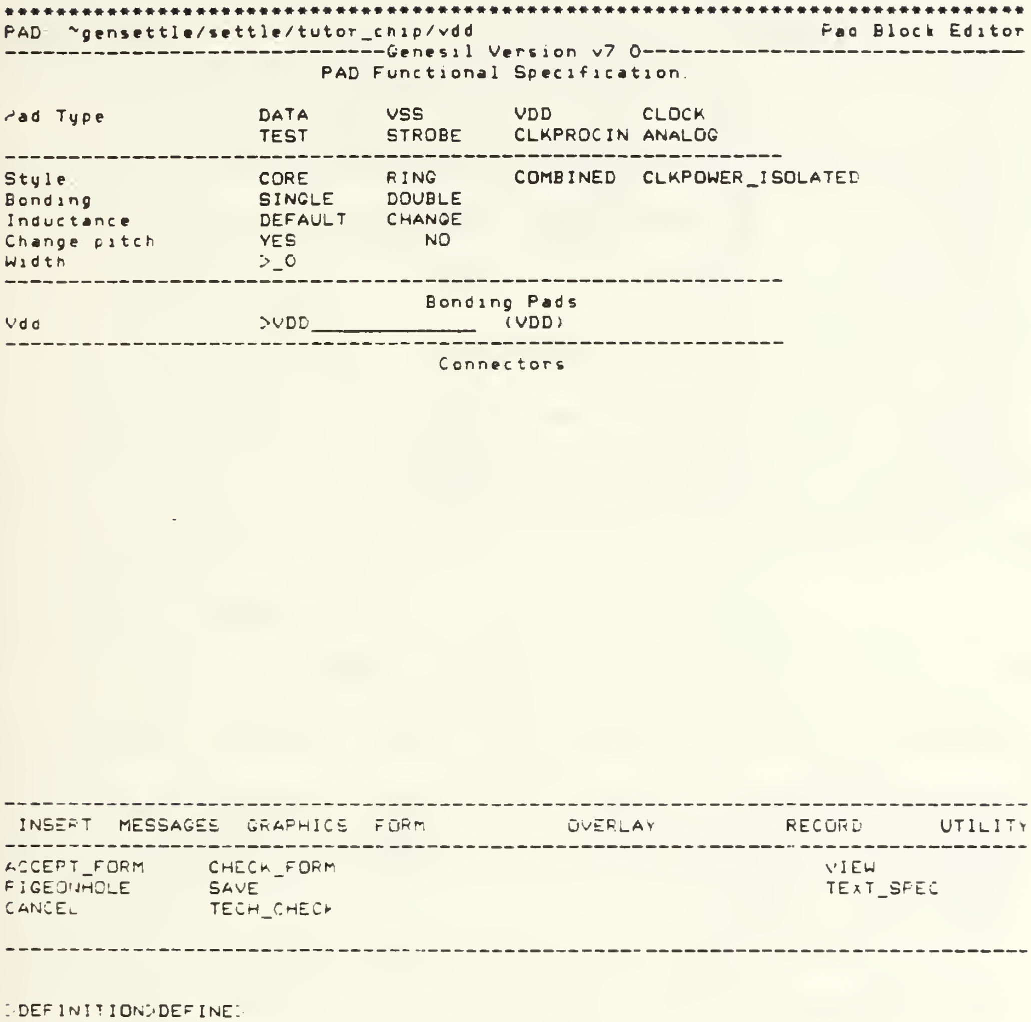
```
*********************************************************************************
PAD  ~gensettle/settle/tutor_chip/vss                              Pad Block Editor
-------------------------------Genesil Version v7.0------------------------------
                              PAD Functional Specification:

Pad Type                 DATA       VSS        VDD         CLOCK
                         TEST       STROBE     CLKPROCIN   ANALOG
-------------------------------------------------------------
Style                    CORE       RING       COMBINED    CLKPOWER_ISOLATED
Bonding                  SINGLE     DOUBLE
Inductance               DEFAULT    CHANGE
Change pitch             YES          NO
Width                    >_0
-------------------------------------------------------------
                                       Bonding Pads:
Vss                      >VSS_____________   (VSS)
-------------------------------------------------------------
                                        Connectors










---------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY          RECORD      UTILITY
---------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_FORM                                      VIEW
PIGEONHOLE      SAVE                                            TEXT_SPEC
CANCEL          TECH_CHECK

---------------------------------------------------------------------------------

>DEFINITION>DEFINE>
```

Figure A.58



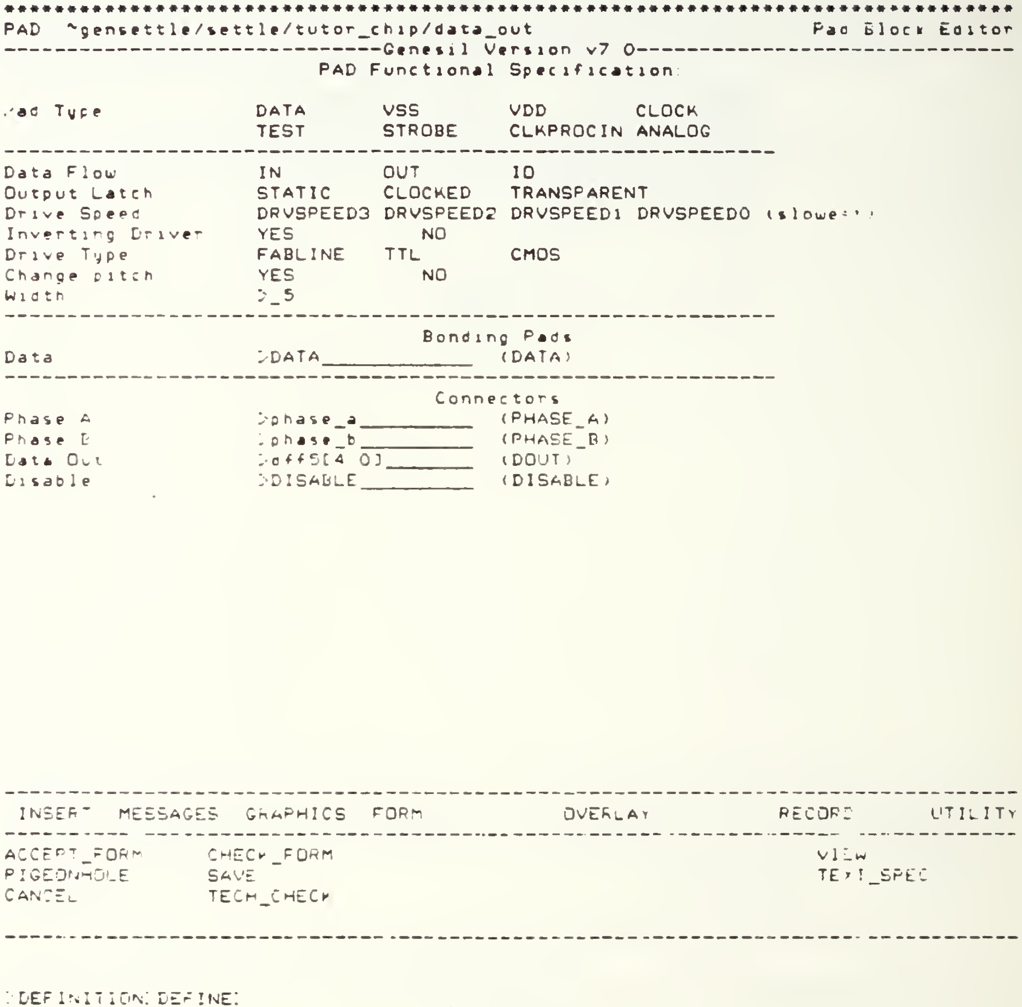
```
**********************************************************************************
Chip  ^gensettle/settle/tutor_chip                                      Definition
-----------------------------Genesil Version v7 0-------------------------------
Sub-Objects Name  Type
 data_in_________ PAD_________________               DEFINE DETACH
 clock___________ PAD_________________               DEFINE DETACH
 vss_____________ PAD_________________               DEFINE DETACH
--------------------------------------------------------------------------------
ATTACH_NEW                            ATTACH_EXISTING
--------------------------------------------------------------------------------








--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
CANCEL                             DEFAULT_TO_CURRENT_NAME


--------------------------------------------------------------------------------

Enter New Name of Created Object [path]
>DEFINITION>ATTACH>ATTACH_EXISTING>^gensettle/settle/tutorblk_1>
```

Figure A.59



```
********************************************************************************
PAD  ~gensettle/settle/tutor_chip/vdd                            Pad Block Editor
-------------------------------Genesil Version v7.0-------------------------------
                         PAD Functional Specification.

Pad Type                DATA       VSS        VDD        CLOCK
                        TEST       STROBE     CLKPROCIN  ANALOG
-----------------------------------------------------------------
Style                   CORE       RING       COMBINED   CLKPOWER_ISOLATED
Bonding                 SINGLE     DOUBLE
Inductance              DEFAULT    CHANGE
Change pitch            YES           NO
Width                   >_0
-----------------------------------------------------------------
                                  Bonding Pads
Vdd                     >VDD_____________  (VDD)
-----------------------------------------------------------------
                                   Connectors









--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_FORM                                       VIEW
PIGEONHOLE       SAVE                                             TEXT_SPEC
CANCEL           TECH_CHECK

--------------------------------------------------------------------------------

>DEFINITION>DEFINE>
```

Figure A.60



```
********************************************************************************
PAD  ~gensettle/settle/tutor_chip/data_out                           Pad Block Editor
------------------------------Genesil Version v7 0------------------------------
                              PAD Functional Specification

Pad Type                DATA      VSS       VDD       CLOCK
                        TEST      STROBE    CLKPROCIN ANALOG
-------------------------------------------------------------
Data Flow               IN        OUT       IO
Output Latch            STATIC    CLOCKED   TRANSPARENT
Drive Speed             DRVSPEED3 DRVSPEED2 DRVSPEED1 DRVSPEED0 (slowest)
Inverting Driver        YES          NO
Drive Type              FABLINE   TTL       CMOS
Change pitch            YES          NO
Width                   >_5
-------------------------------------------------------------
                                 Bonding Pads
Data                    >DATA___________  (DATA)
-------------------------------------------------------------
                                  Connectors
Phase A                 >phase_a________  (PHASE_A)
Phase B                 >phase_b________  (PHASE_B)
Data Out                >dffS[4 0]______  (DOUT)
Disable                 >DISABLE________  (DISABLE)








--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD       UTILITY
---------- ---------------------------------------------------------- ------------
ACCEPT_FORM      CHECK_FORM                                      VIEW
PIGEONHOLE       SAVE                                            TEXT_SPEC
CANCEL           TECH_CHECK

--------------------------------------------------------------------------------

>DEFINITION: DEFINE:
```

Figure A.61

m. From the Chip Specification form, select ATTACH_EXISTING. The user will now be prompted for the path to the existing object. Type in ~gensettle/settle/tutorblk_1 and depress RETURN (use your own name and path unless you are in Settle's account). Now select DEFAULT_TO_CURRENT_NAME. The Chip Specification form should now include tutorblk_1.

n. Attach tutorblk_2 in the same manner. The Chip Specification form should now contain the same objects as Figure A.56.

2. The Chip must now be netlisted:

a. Select OBJECT_NETLIST (Figure A.56) and select each object's netlist, as described in the module section and ensure all object netlists match Figure A.62 - A.67. Type in the required signal names where applicable, and depress RETURN at the end of each line.

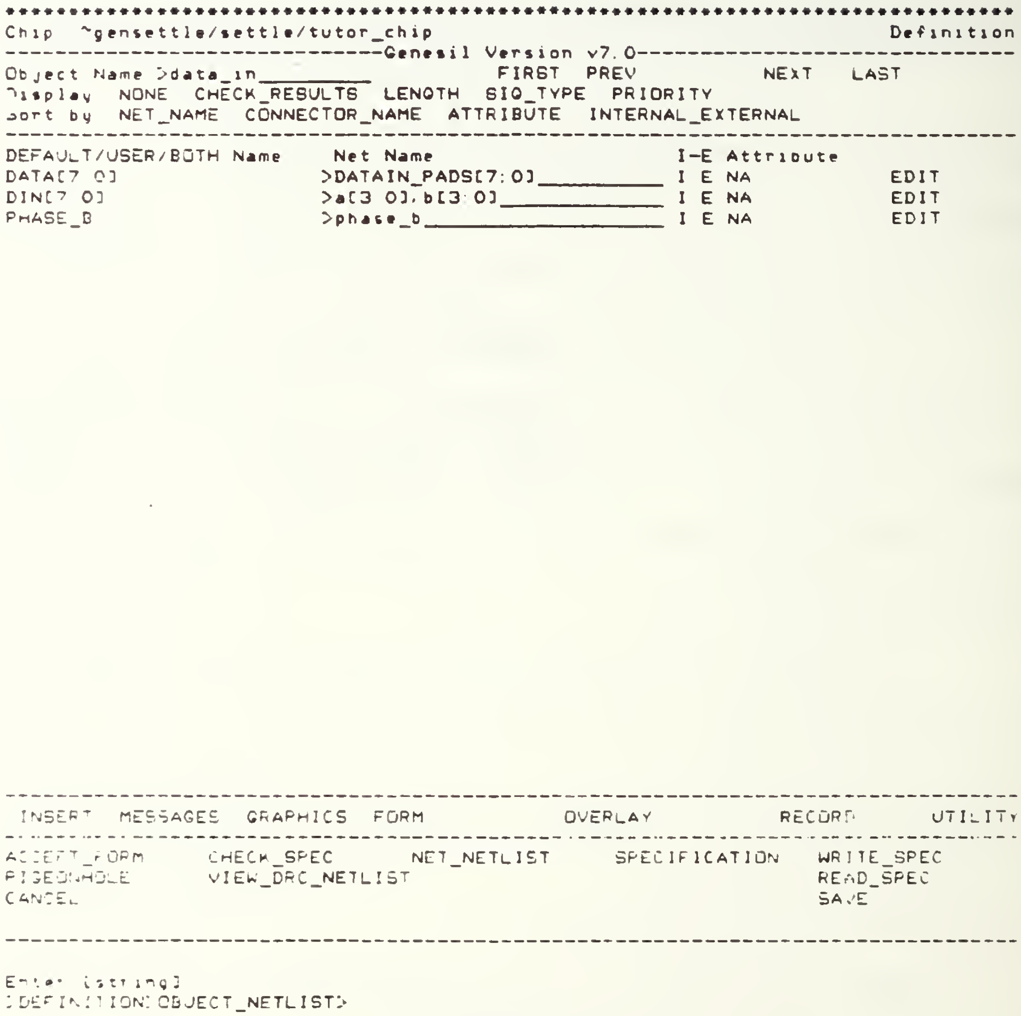
b. Select SPECIFICATION (Figure A.67), and if there are no netlist errors, the Chip Specification form should be on the screen (Figure A.56). Now select ACCEPT_FORM (Figure A.56), and the screen will now show the Definition menu as illustrated in Figure A.68.

3. The Chip must now be floorplanned:

a. Select FLOORPLAN (Figure A.68).

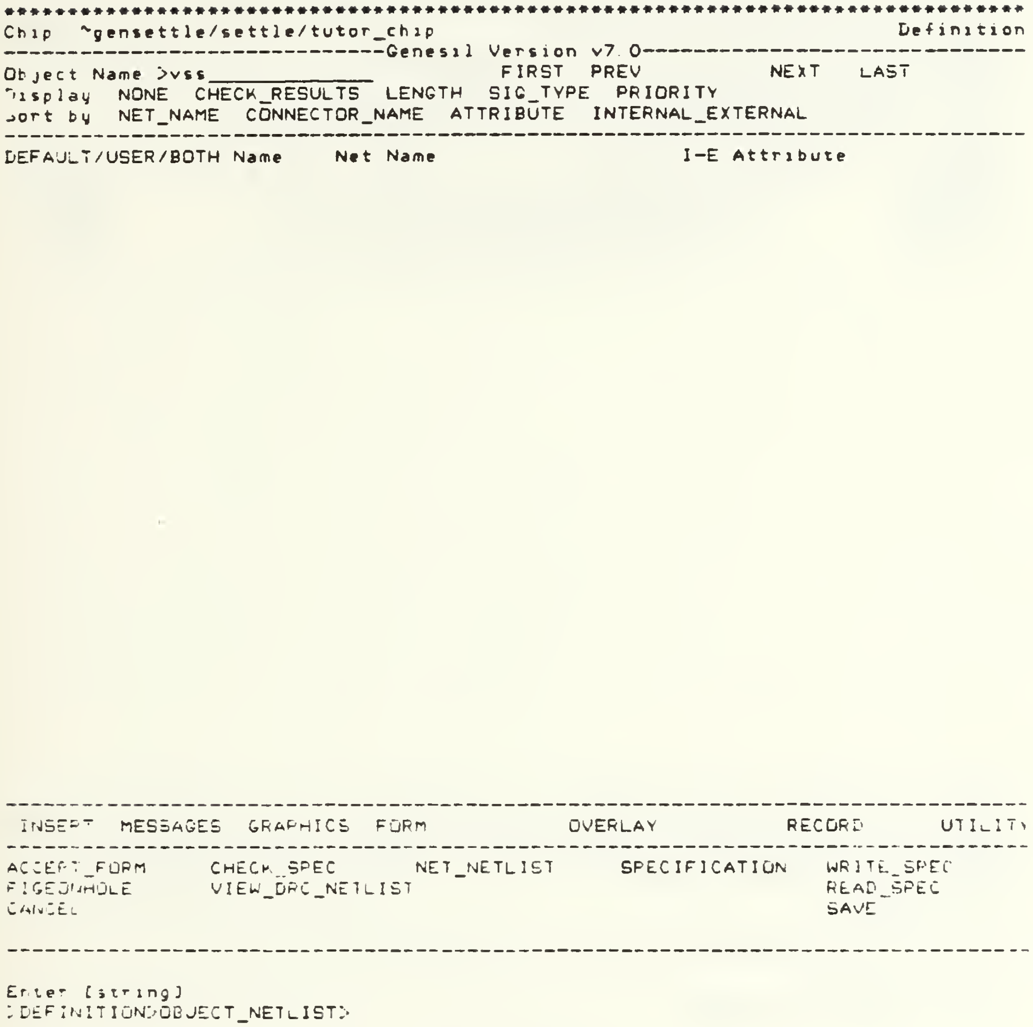
b. Use the module placement procedures to place the two unplaced blocks (Figure A.69).

c. Go BACK to the initial floorplan menu (Figure A.70). Select PINOUT. All unplaced PADS are now listed.



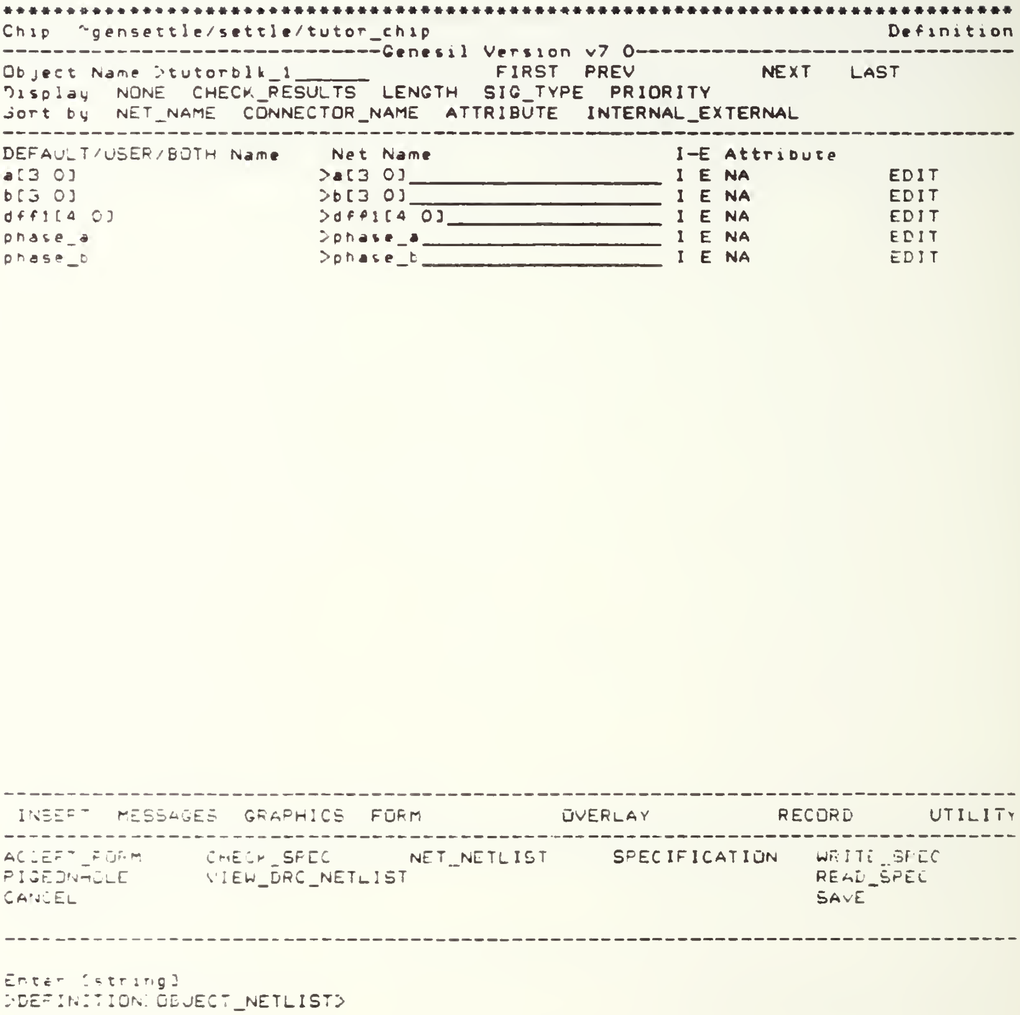
```
******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                  Definition
------------------------------Genesil Version v7.0-----------------------------
Object Name >data_in_________          FIRST  PREV         NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name     Net Name                     I-E Attribute
DATA[7 0]                 >DATAIN_PADS[7:0]__________   I E NA          EDIT
DIN[7 0]                  >a[3 0],b[3:0]_____________   I E NA          EDIT
PHASE_B                   >phase_b___________________   I E NA          EDIT













------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM              OVERLAY          RECORD     UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC       NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                                 READ_SPEC
CANCEL                                                           SAVE

------------------------------------------------------------------------------

Enter [string]
[DEFINITION]OBJECT_NETLIST>
```

Figure A.62



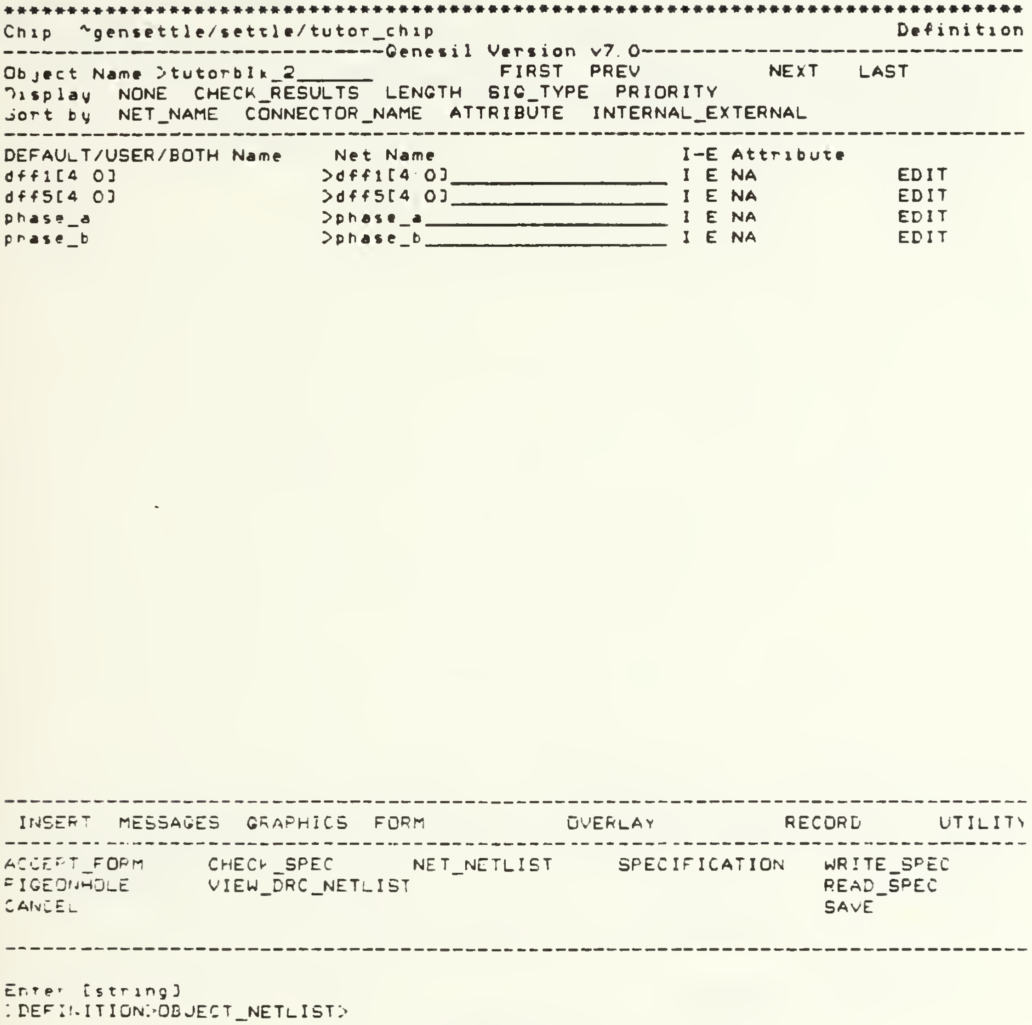
```
********************************************************************************
Chip  ~gensettle/settle/tutor_chip                                    Definition
------------------------------Genesil Version v7.0------------------------------
Object Name >vss_____________          FIRST  PREV          NEXT    LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
--------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                    I-E Attribute


















--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY           RECORD      UTILITY
--------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_SPEC       NET_NETLIST      SPECIFICATION    WRITE_SPEC
PIGEONHOLE       VIEW_DRC_NETLIST                                   READ_SPEC
CANCEL                                                              SAVE

--------------------------------------------------------------------------------

Enter [string]
>DEFINITION>OBJECT_NETLIST>
```

Figure A.63



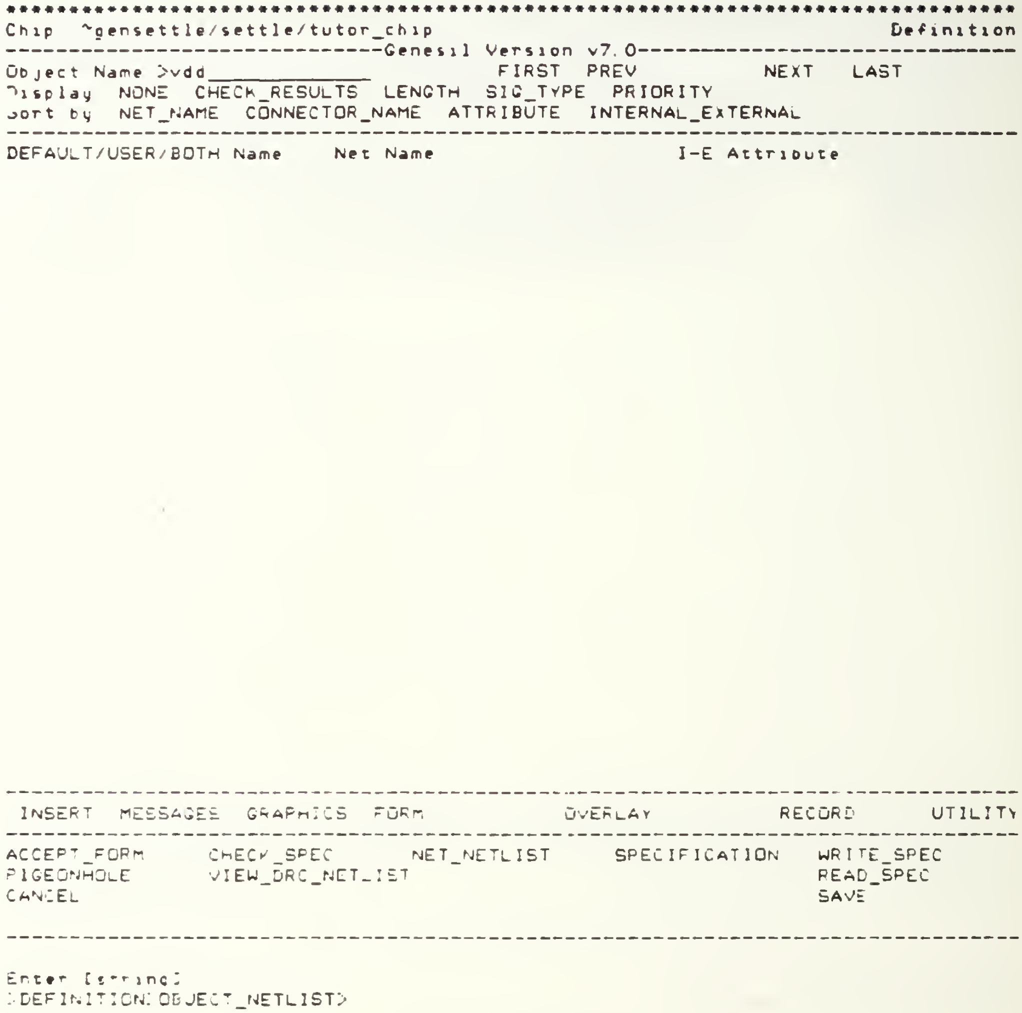
```
******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                 Definition
-----------------------------Genesil Version v7 0-----------------------------
Object Name >tutorblk_1______          FIRST  PREV           NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name     Net Name                    I-E Attribute
a[3 0]                    >a[3 0]___________________  I E NA             EDIT
b[3 0]                    >b[3 0]___________________  I E NA             EDIT
dff1[4 0]                 >dff1[4 0]________________  I E NA             EDIT
phase_a                   >phase_a__________________  I E NA             EDIT
phase_b                   >phase_b__________________  I E NA             EDIT

------------------------------------------------------------------------------
 INSERT   MESSAGES   GRAPHICS   FORM            OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC      NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                                READ_SPEC
CANCEL                                                          SAVE

------------------------------------------------------------------------------

Enter [string]
>DEFINITION:OBJECT_NETLIST>
```

Figure A.64



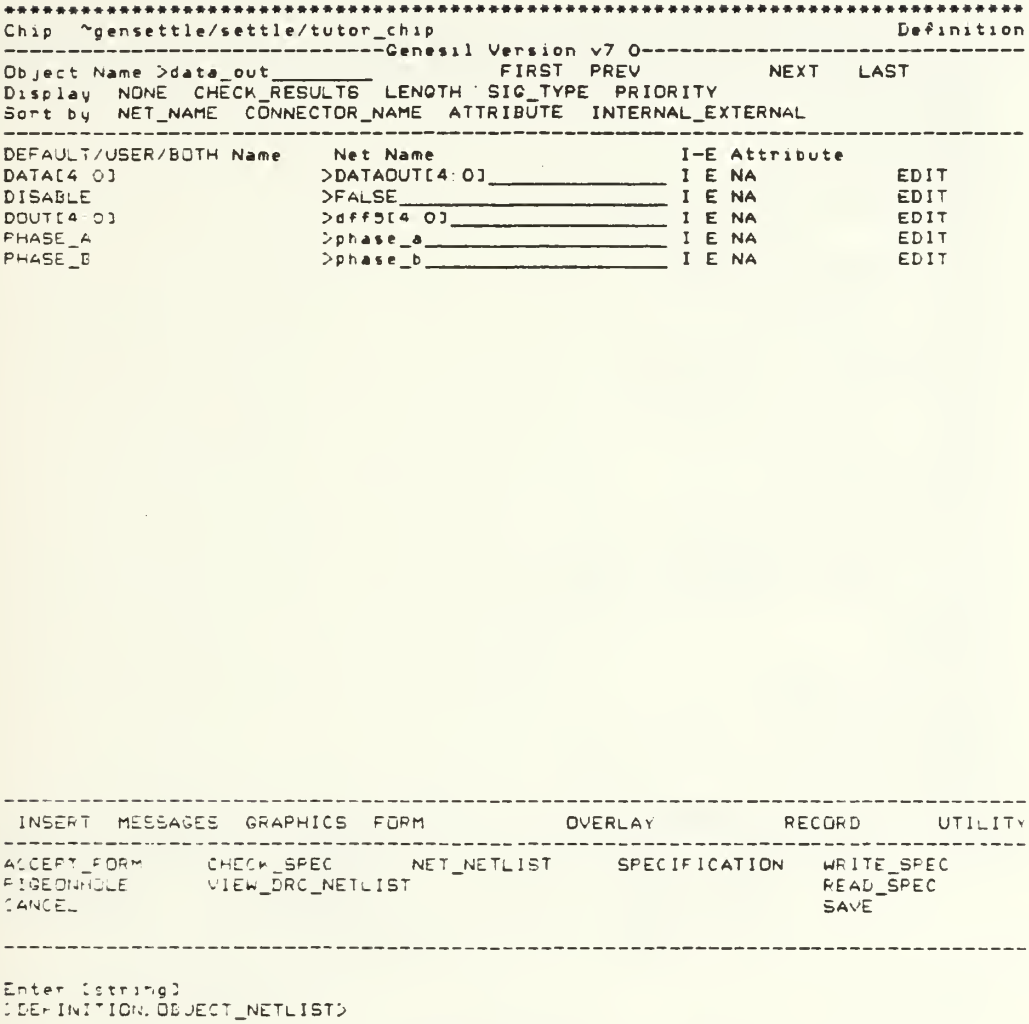
```
*****************************************************************************
Chip  ^gensettle/settle/tutor_chip                                 Definition
-----------------------------Genesil Version v7.0-----------------------------
Object Name >tutorblk_2______         FIRST  PREV           NEXT    LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
-----------------------------------------------------------------------------
DEFAULT/USER/BOTH Name     Net Name                   I-E Attribute
dff1[4 0]                 >dff1[4 0]________________  I E NA             EDIT
dff5[4 0]                 >dff5[4 0]________________  I E NA             EDIT
phase_a                   >phase_a__________________  I E NA             EDIT
phase_b                   >phase_b__________________  I E NA             EDIT




-----------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM            OVERLAY          RECORD     UTILITY
-----------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC      NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                                READ_SPEC
CANCEL                                                          SAVE

-----------------------------------------------------------------------------

Enter [string]
[DEFINITION>OBJECT_NETLIST>
```

Figure A.65



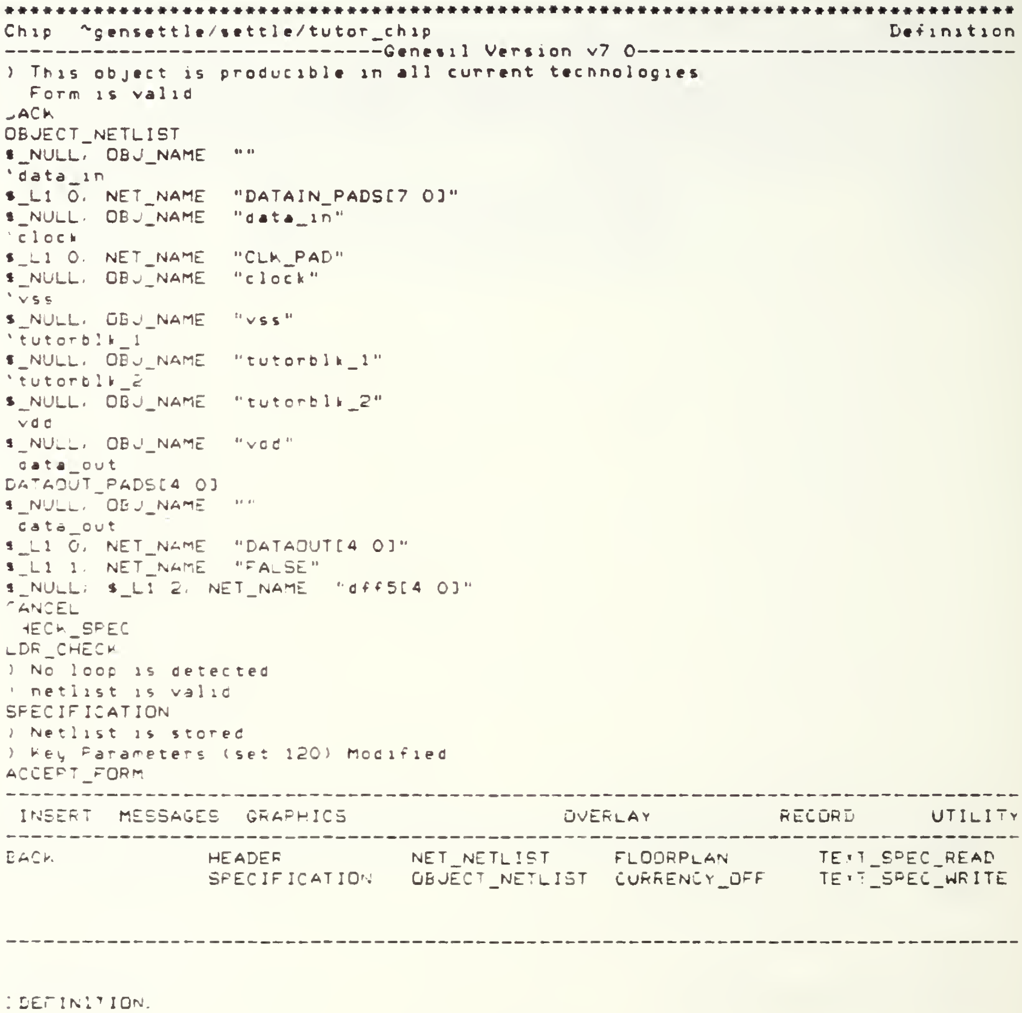
```
*******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                   Definition
----------------------------Genesil Version v7.0------------------------------
Object Name >vdd_______________          FIRST  PREV          NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
-------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                    I-E Attribute











-------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY          RECORD      UTILITY
-------------------------------------------------------------------------------
ACCEPT_FORM      CHECK_SPEC       NET_NETLIST      SPECIFICATION    WRITE_SPEC
PIGEONHOLE       VIEW_DRC_NET_LIST                                  READ_SPEC
CANCEL                                                              SAVE

-------------------------------------------------------------------------------

Enter [string]
:DEFINITION:OBJECT_NETLIST>
```

Figure A.66



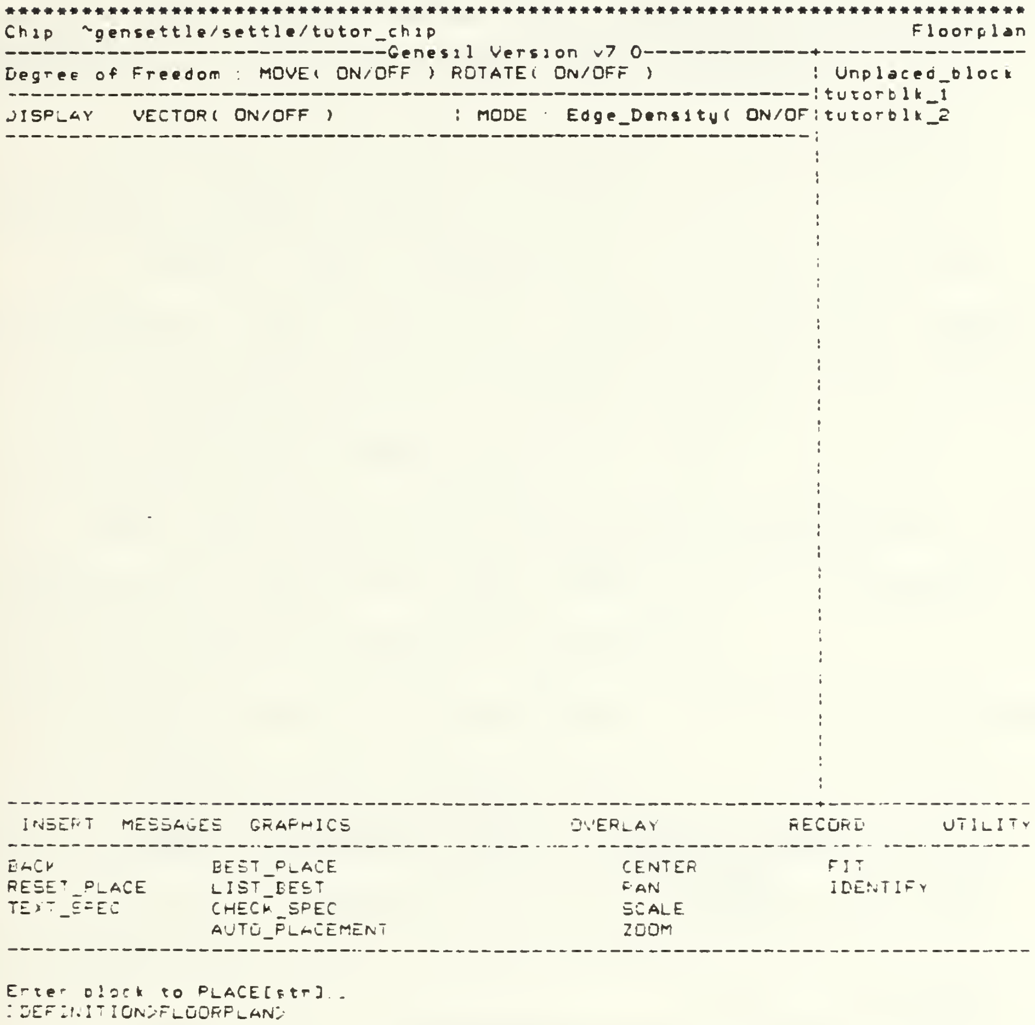
```
******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                  Definition
-----------------------------Genesil Version v7 0-----------------------------
Object Name >data_out________           FIRST  PREV          NEXT   LAST
Display  NONE  CHECK_RESULTS  LENGTH  SIG_TYPE  PRIORITY
Sort by  NET_NAME  CONNECTOR_NAME  ATTRIBUTE  INTERNAL_EXTERNAL
------------------------------------------------------------------------------
DEFAULT/USER/BOTH Name    Net Name                    I-E Attribute
DATA[4 0]                >DATAOUT[4:0]_____________   I E NA           EDIT
DISABLE                  >FALSE____________________   I E NA           EDIT
DOUT[4 0]                >dff5[4 0]________________   I E NA           EDIT
PHASE_A                  >phase_a__________________   I E NA           EDIT
PHASE_B                  >phase_b__________________   I E NA           EDIT












------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM           OVERLAY          RECORD      UTILITY
------------------------------------------------------------------------------
ACCEPT_FORM     CHECK_SPEC     NET_NETLIST     SPECIFICATION   WRITE_SPEC
PIGEONHOLE      VIEW_DRC_NETLIST                               READ_SPEC
CANCEL                                                         SAVE

------------------------------------------------------------------------------

Enter [string]
[DEFINITION.OBJECT_NETLIST>
```

Figure A.67



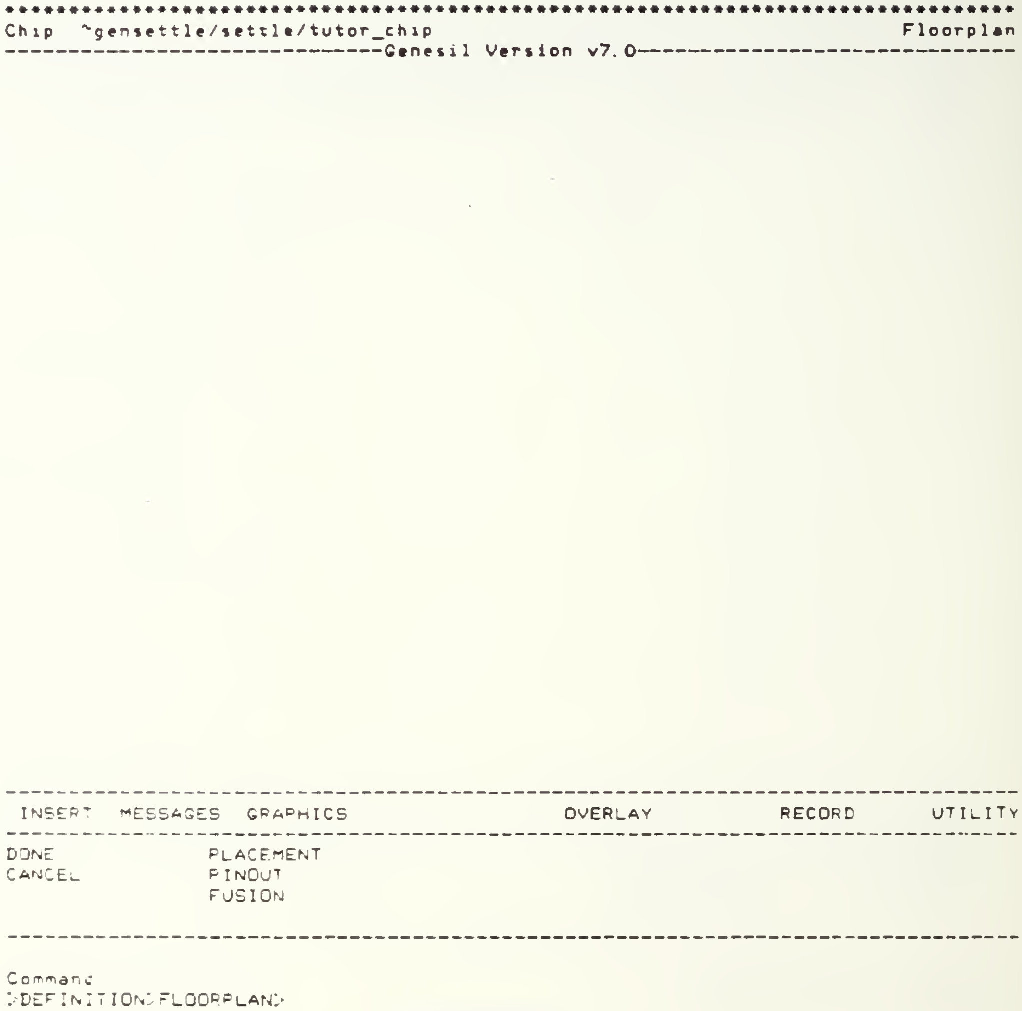
```
*******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                   Definition
------------------------------Genesil Version v7 0-----------------------------
) This object is producible in all current technologies
  Form is valid
BACK
OBJECT_NETLIST
$_NULL, OBJ_NAME  ""
'data_in
$_L1 0, NET_NAME  "DATAIN_PADS[7 0]"
$_NULL, OBJ_NAME  "data_in"
'clock
$_L1 0, NET_NAME  "CLK_PAD"
$_NULL, OBJ_NAME  "clock"
'vss
$_NULL, OBJ_NAME  "vss"
'tutorblk_1
$_NULL, OBJ_NAME  "tutorblk_1"
'tutorblk_2
$_NULL, OBJ_NAME  "tutorblk_2"
 vdd
$_NULL, OBJ_NAME  "vdd"
 data_out
DATAOUT_PADS[4 0]
$_NULL, OBJ_NAME  ""
 data_out
$_L1 0, NET_NAME  "DATAOUT[4 0]"
$_L1 1, NET_NAME  "FALSE"
$_NULL; $_L1 2, NET_NAME  "dff5[4 0]"
CANCEL
 HECK_SPEC
LDR_CHECK
) No loop is detected
' netlist is valid
SPECIFICATION
) Netlist is stored
) Key Parameters (set 120) Modified
ACCEPT_FORM
-------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                    OVERLAY          RECORD     UTILITY
-------------------------------------------------------------------------------
BACK           HEADER          NET_NETLIST     FLOORPLAN       TEXT_SPEC_READ
               SPECIFICATION   OBJECT_NETLIST  CURRENCY_OFF    TEXT_SPEC_WRITE


-------------------------------------------------------------------------------

:DEFINITION.
```

Figure A.68



```
*******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                    Floorplan
-----------------------------Genesil Version v7 0--------------+----------------
Degree of Freedom : MOVE( ON/OFF ) ROTATE( ON/OFF )            : Unplaced_block
---------------------------------------------------------------:tutorblk_1
DISPLAY   VECTOR( ON/OFF )         : MODE   Edge_Density( ON/OF:tutorblk_2
---------------------------------------------------------------:
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
                                                               :
---------------------------------------------------------------+----------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY         RECORD      UTILITY
-------------------------------------------------------------------------------
BACK            BEST_PLACE                      CENTER          FIT
RESET_PLACE     LIST_BEST                       PAN             IDENTIFY
TEXT_SPEC       CHECK_SPEC                      SCALE
                AUTO_PLACEMENT                  ZOOM
-------------------------------------------------------------------------------

Enter block to PLACE[str]..
:DEFINITION>FLOORPLAN>
```

Figure A.69



```
********************************************************************************
Chip  ~gensettle/settle/tutor_chip                                     Floorplan
------------------------------Genesil Version v7.0------------------------------





















--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY           RECORD     UTILITY
--------------------------------------------------------------------------------
DONE            PLACEMENT
CANCEL          PINOUT
                FUSION

--------------------------------------------------------------------------------

Command
>DEFINITION>FLOORPLAN>
```

Figure A.70

Select each PAD with the MOUSE. Place the cross hairs around the chip in the desired PAD location. Click the right MOUSE button. After placing a PAD, it may be moved by hooking it with the MOUSE, and moving it to the desired new location.

d. After placing all PADS, go BACK to the initial floorplan menu (Figure A.70) and select FUSION.

e. Select AUTO_FUSE. Go BACK to the initial floorplan menu (Figure A.70).

f. Select DONE (Figure A.70). If all pads are placed correctly, the system will confirm floorplan complete.

g. If there are PAD placement errors, try moving the PADS around and select DONE again.

4. After the chip is netlisted and floorplanned, it may be simulated and timing analysis performed in accordance with the previous block and module instructions.

5. The chip may be plotted using the following commands:

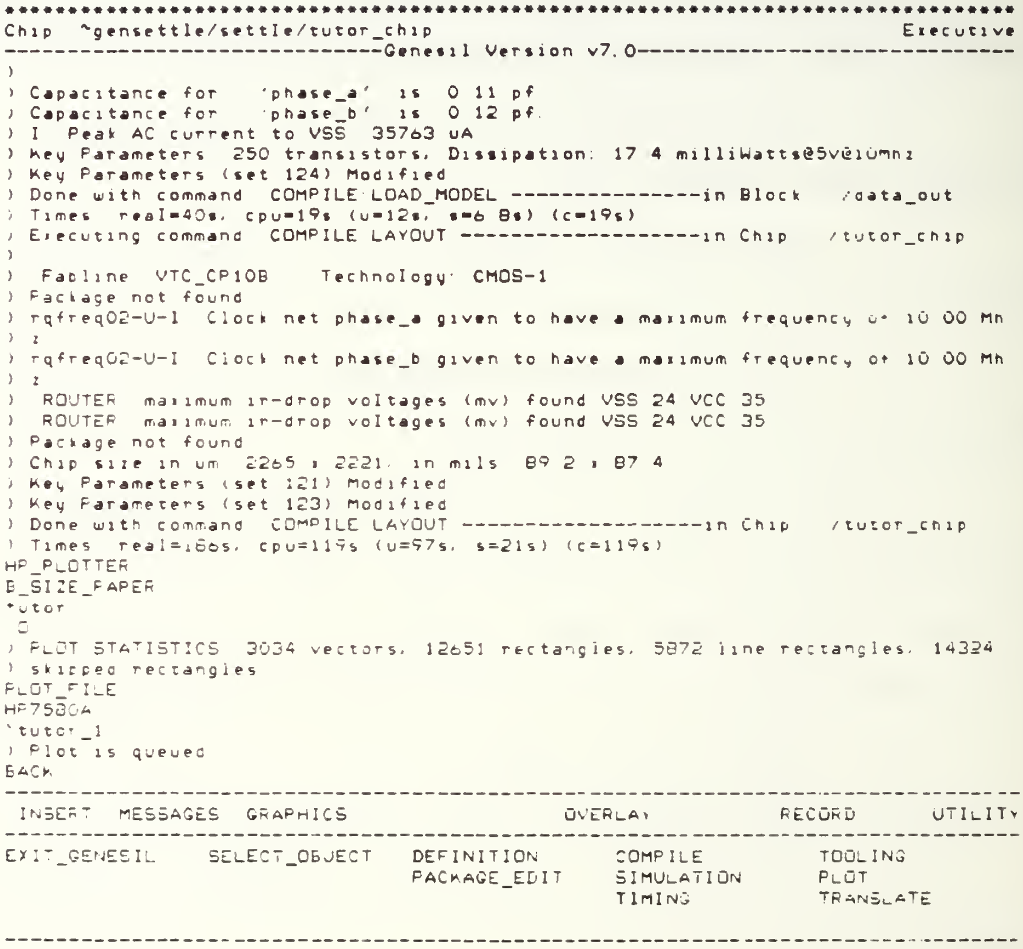
a. Select PLOT (Figure A.71).

b. Next select NEW_PLOT (Figure A.72).

c. Select LAYOUT for a VLSI layout of the chip (Figure A.73).

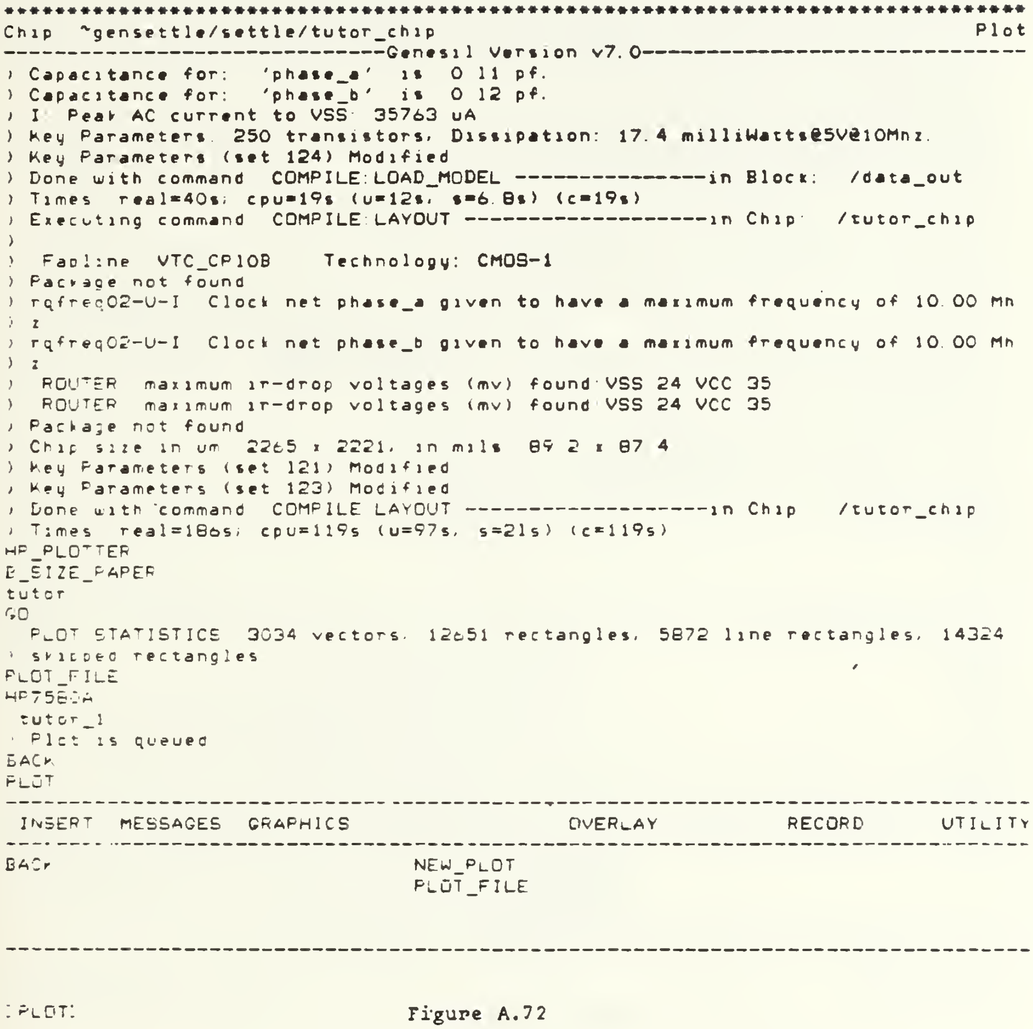
d. Now select WORKSTATION (Figure A.74), and then GO (Figure A.75).

e. If all things go well, the screen should show a layout similar to Figure A.76.



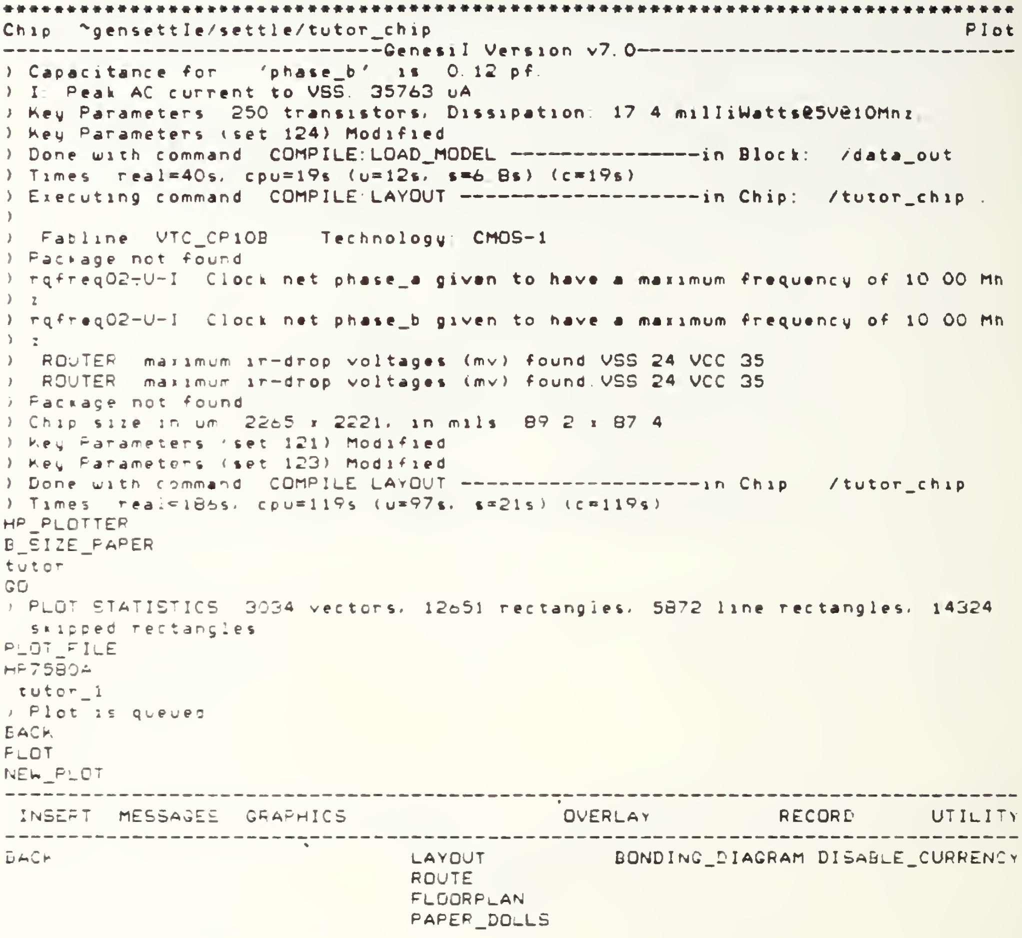
```
*******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                      Executive
-----------------------------Genesil Version v7.0-------------------------------
)
) Capacitance for   'phase_a'  is  0 11 pf
) Capacitance for   'phase_b'  is  0 12 pf.
) I  Peak AC current to VSS  35763 uA
) Key Parameters  250 transistors, Dissipation: 17 4 milliWatts@5v@10mhz
) Key Parameters (set 124) Modified
) Done with command  COMPILE LOAD_MODEL ---------------in Block   /data_out
) Times  real=40s, cpu=19s (u=12s, s=6 8s) (c=19s)
) Executing command  COMPILE LAYOUT --------------------in Chip   /tutor_chip
)
)  Fabline  VTC_CP10B    Technology: CMOS-1
) Package not found
) rqfreq02-U-I  Clock net phase_a given to have a maximum frequency of 10 00 Mh
) z
) rqfreq02-U-I  Clock net phase_b given to have a maximum frequency of 10 00 Mh
) z
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
) Package not found
) Chip size in um  2265 x 2221  in mils  89 2 x 87 4
) Key Parameters (set 121) Modified
) Key Parameters (set 123) Modified
) Done with command  COMPILE LAYOUT --------------------in Chip   /tutor_chip
) Times  real=186s, cpu=119s (u=97s, s=21s) (c=119s)
HP_PLOTTER
B_SIZE_PAPER
tutor
 0
) PLOT STATISTICS  3034 vectors, 12651 rectangles, 5872 line rectangles, 14324
) skipped rectangles
PLOT_FILE
HP7580A
`tutor_1
) Plot is queued
BACK
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY            RECORD       UTILITY
--------------------------------------------------------------------------------
EXIT_GENESIL    SELECT_OBJECT    DEFINITION       COMPILE          TOOLING
                                 PACKAGE_EDIT     SIMULATION       PLOT
                                                  TIMING           TRANSLATE

--------------------------------------------------------------------------------
```

Figure A.71



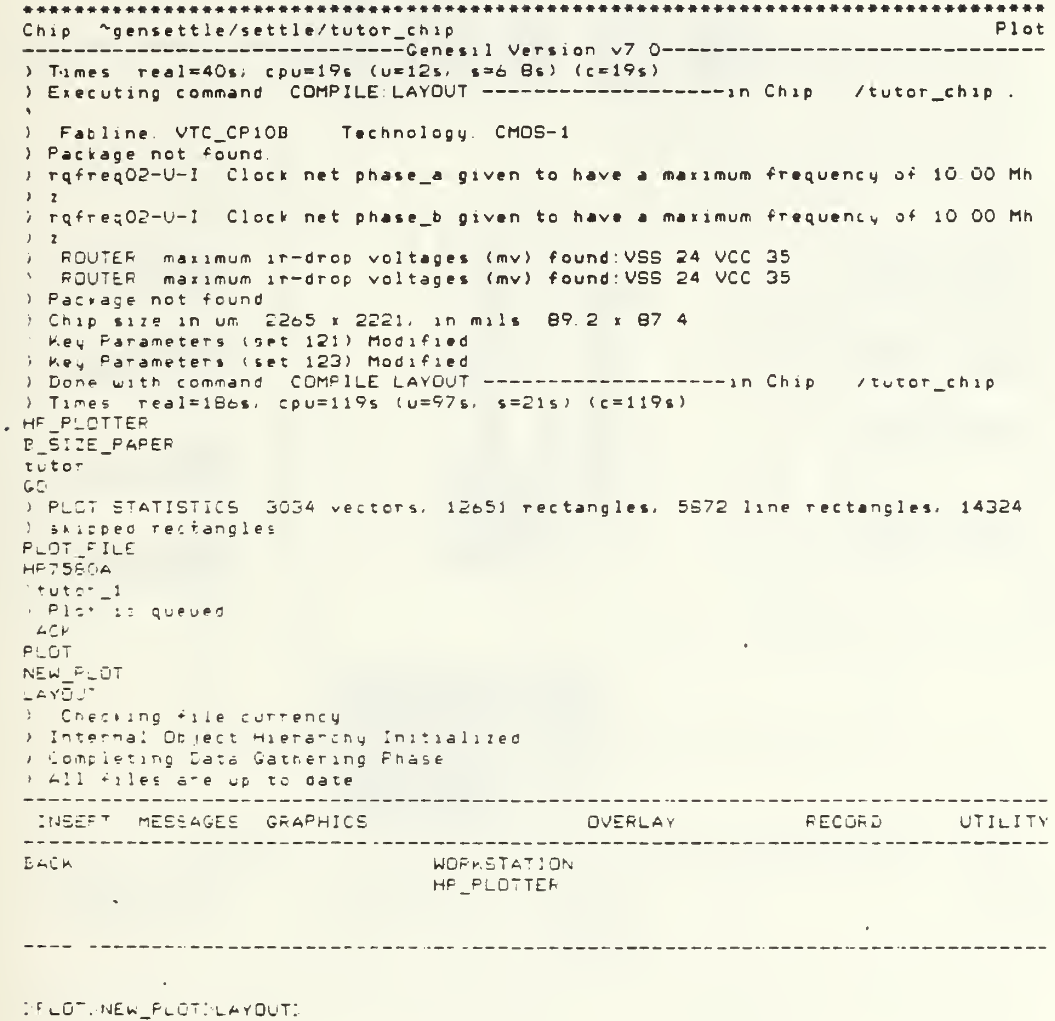
```
********************************************************************************
Chip  ~gensettle/settle/tutor_chip                                          Plot
------------------------------Genesil Version v7.0------------------------------
) Capacitance for:  'phase_a'  is  0 11 pf.
) Capacitance for:  'phase_b'  is  0 12 pf.
) I  Peak AC current to VSS  35763 uA
) Key Parameters  250 transistors, Dissipation: 17.4 milliWatts@5V@10Mhz.
) Key Parameters (set 124) Modified
) Done with command  COMPILE:LOAD_MODEL ---------------in Block:  /data_out
) Times  real=40s; cpu=19s (u=12s, s=6.8s) (c=19s)
) Executing command  COMPILE:LAYOUT -------------------in Chip   /tutor_chip
)
)  Fabline  VTC_CP10B    Technology: CMOS-1
) Package not found
) rqfreq02-U-I  Clock net phase_a given to have a maximum frequency of 10.00 Mh
) z
) rqfreq02-U-I  Clock net phase_b given to have a maximum frequency of 10.00 Mh
) z
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
) Package not found
) Chip size in um  2265 x 2221, in mils  89 2 x 87 4
) Key Parameters (set 121) Modified
) Key Parameters (set 123) Modified
) Done with command  COMPILE LAYOUT -------------------in Chip   /tutor_chip
) Times  real=186s; cpu=119s (u=97s, s=21s) (c=119s)
HP_PLOTTER
B_SIZE_PAPER
tutor
GO
  PLOT STATISTICS  3034 vectors. 12651 rectangles. 5872 line rectangles. 14324
) skipped rectangles
PLOT_FILE
HP7580A
 tutor_1
) Plot is queued
BACK
PLOT
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                 OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK                                NEW_PLOT
                                    PLOT_FILE


--------------------------------------------------------------------------------

[PLOT]
```

Figure A.72



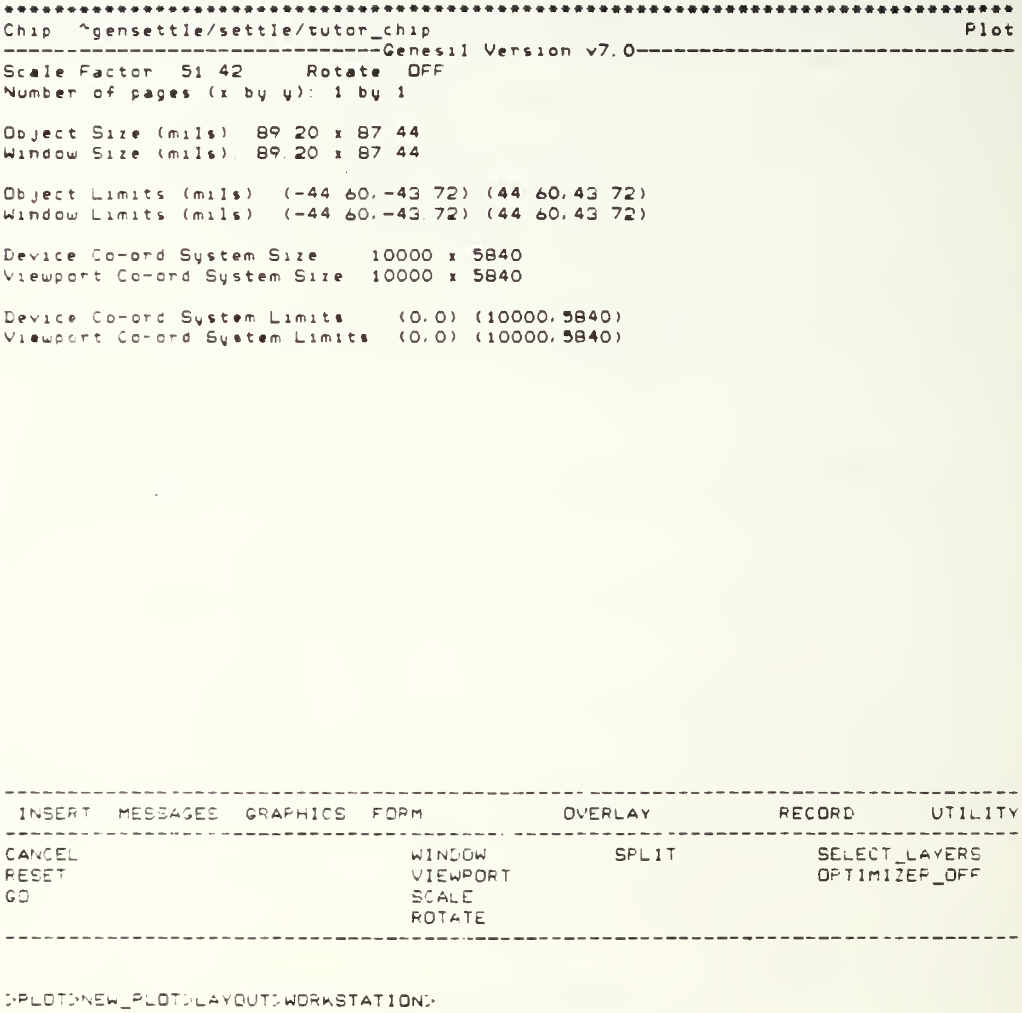
```
******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                         Plot
------------------------------Genesil Version v7.0------------------------------
) Capacitance for   'phase_b'  is  0.12 pf.
) I: Peak AC current to VSS. 35763 uA
) Key Parameters  250 transistors, Dissipation: 17.4 milliWatts@5V@10Mhz
) Key Parameters (set 124) Modified
) Done with command  COMPILE:LOAD_MODEL ----------------in Block:  /data_out
) Times  real=40s, cpu=19s (u=12s, s=6.8s) (c=19s)
) Executing command  COMPILE:LAYOUT --------------------in Chip:  /tutor_chip .
)
)  Fabline  VTC_CP10B    Technology: CMOS-1
) Package not found
) rqfreq02-U-I  Clock net phase_a given to have a maximum frequency of 10.00 Mh
) z
) rqfreq02-U-I  Clock net phase_b given to have a maximum frequency of 10.00 Mh
) z
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
)  ROUTER  maximum ir-drop voltages (mv) found VSS 24 VCC 35
) Package not found
) Chip size in um  2265 x 2221, in mils  89.2 x 87.4
) Key Parameters (set 121) Modified
) Key Parameters (set 123) Modified
) Done with command  COMPILE:LAYOUT --------------------in Chip:  /tutor_chip
) Times  real=186s, cpu=119s (u=97s, s=21s) (c=119s)
HP_PLOTTER
B_SIZE_PAPER
tutor
GO
) PLOT STATISTICS  3034 vectors, 12651 rectangles, 5872 line rectangles, 14324
  skipped rectangles
PLOT_FILE
HP7580A
 tutor_1
) Plot is queued
BACK
PLOT
NEW_PLOT
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                     OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK                           LAYOUT          BONDING_DIAGRAM DISABLE_CURRENCY
                               ROUTE
                               FLOORPLAN
                               PAPER_DOLLS
--------------------------------------------------------------------------------

[PLOT]-NEW_PLOT:
```

Figure A.73



```
*******************************************************************************
Chip  ~gensettle/settle/tutor_chip                                         Plot
------------------------------Genesil Version v7 0------------------------------
) Times  real=40s; cpu=19s (u=12s, s=6 8s) (c=19s)
) Executing command  COMPILE:LAYOUT -------------------in Chip   /tutor_chip .
,
)  Fabline. VTC_CP10B    Technology. CMOS-1
) Package not found.
) rqfreq02-U-I  Clock net phase_a given to have a maximum frequency of 10.00 Mh
) z
) rqfreq02-U-I  Clock net phase_b given to have a maximum frequency of 10 00 Mh
) z
)  ROUTER  maximum ir-drop voltages (mv) found:VSS 24 VCC 35
)  ROUTER  maximum ir-drop voltages (mv) found:VSS 24 VCC 35
) Package not found
) Chip size in um  2265 x 2221, in mils  89.2 x 87 4
) Key Parameters (set 121) Modified
) Key Parameters (set 123) Modified
) Done with command  COMPILE LAYOUT -------------------in Chip   /tutor_chip
) Times  real=186s, cpu=119s (u=97s, s=21s) (c=119s)
HP_PLOTTER
B_SIZE_PAPER
tutor
GO
) PLOT STATISTICS  3034 vectors, 12651 rectangles, 5872 line rectangles, 14324
) skipped rectangles
PLOT_FILE
HP7580A
tutor_1
) Plot is queued
 ACK
PLOT
NEW_PLOT
LAYOUT
)  Checking file currency
) Internal Object Hierarchy Initialized
) Completing Data Gathering Phase
) All files are up to date
--------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS                OVERLAY          RECORD      UTILITY
--------------------------------------------------------------------------------
BACK                           WORKSTATION
                               HP_PLOTTER


---- ---------------------------------------------------------------------------


:PLOT:NEW_PLOT:LAYOUT:
```

Figure A.74



```
*****************************************************************************
Chip  ^gensettle/settle/tutor_chip                                       Plot
-----------------------------Genesil Version v7.0----------------------------
Scale Factor  51.42     Rotate  OFF
Number of pages (x by y): 1 by 1

Object Size (mils)  89.20 x 87.44
Window Size (mils)  89.20 x 87.44

Object Limits (mils)  (-44.60,-43.72) (44.60,43.72)
Window Limits (mils)  (-44.60,-43.72) (44.60,43.72)

Device Co-ord System Size     10000 x 5840
Viewport Co-ord System Size   10000 x 5840

Device Co-ord System Limits     (0,0) (10000,5840)
Viewport Co-ord System Limits   (0,0) (10000,5840)




------------------------------------------------------------------------------
 INSERT  MESSAGES  GRAPHICS  FORM          OVERLAY         RECORD      UTILITY
------------------------------------------------------------------------------
CANCEL                          WINDOW        SPLIT           SELECT_LAYERS
RESET                           VIEWPORT                      OPTIMIZER_OFF
GO                              SCALE
                                ROTATE
------------------------------------------------------------------------------

>PLOT>NEW_PLOT>LAYOUT>WORKSTATION>
```

Figure A.75

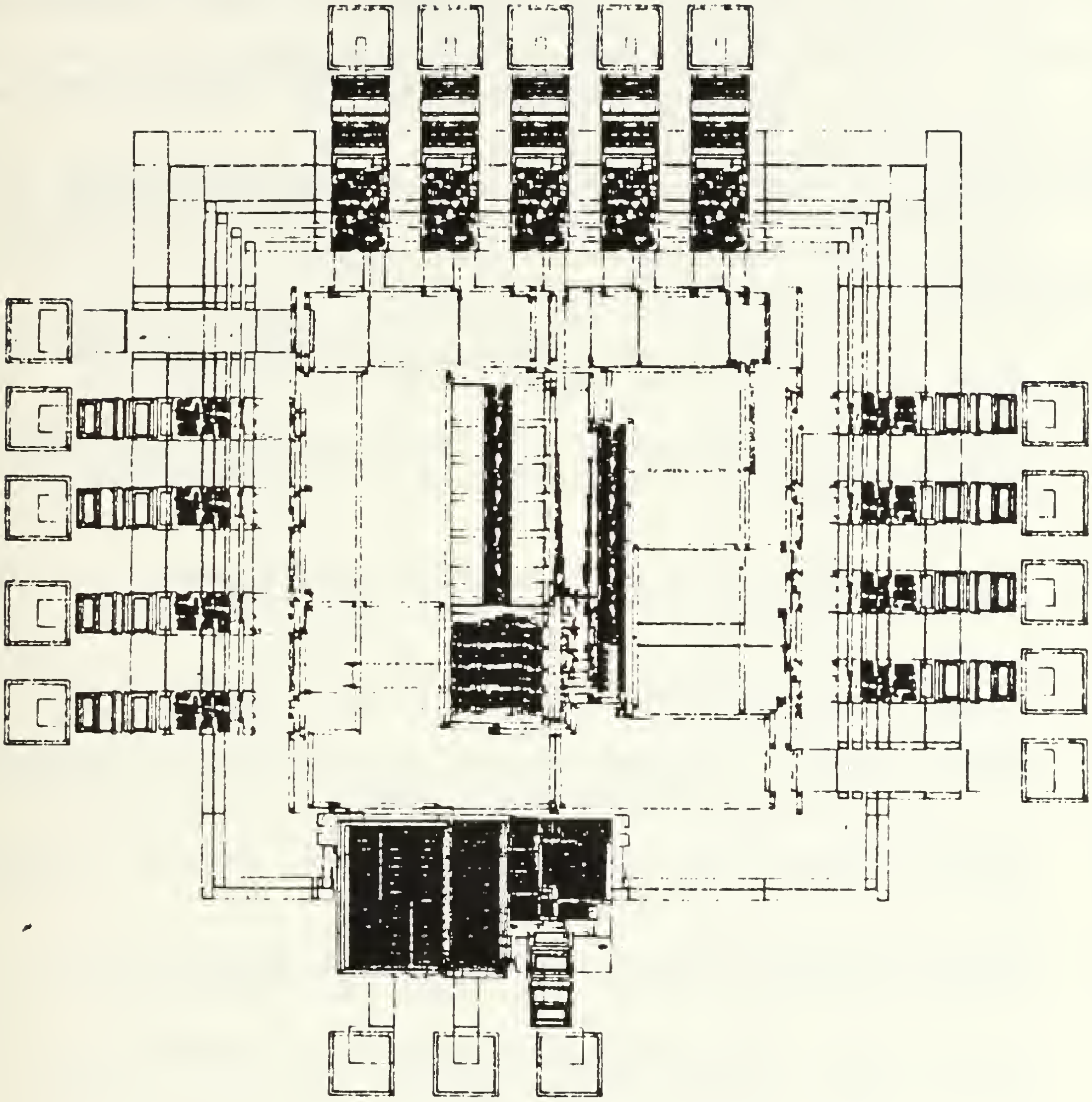


Figure A.76

LIST OF REFERENCES

1. VLSI Systems Design Staff, Directory of Silicon Compilers, VLSI Systems Design, March 1988.

2. Shou, S.H., Silicon Compilation Aids IC Design, Defense Electronics, September 1987.

3. Loomis, H.H. Jr., and Kirk, D.E., "Automated Design of VLSI Devices for Navy Space Applications", Unpublished Paper, Naval Postgraduate School, Monterey, CA, April 1987.

4. Doer, A., and Sabo, D., Silicon Compilation and Design for Test, Solid State Technology, August 1986.

5. Weste, N., and Eshragian, F., Principles of CMOS VLSI Design, A Systems Perspective, p.238, Addison-Wesley, October 1985.

6. Genesil System, System Description Users Manual, Silicon Compiler Systems Corp., San Jose, CA, 1986.

7. Genesil System Compiler Library Volume I, p.1-1, Silicon Compiler Systems Corp., San Jose, CA, 1986.

8. Genesil System, System Description Application Commands, Silicon Compiler Systems Corp., San Jose, CA, 1986.

9. Genesil System, Simulation Users Guide, Silicon Compiler Systems Corp., San Jose, CA, September 1987.

10. Genesil System, Timing Analysis Users Guide, Silicon Compiler Systems Corp., San Jose, CA, July 1987.

11. Stone, H.S., High Performance Computer Architecture, ch. 3, Addison-Wesley Publishing, 1987.

12. Kogge, P.M., The Architecture of Pipelined Computers, Hemisphere Publishing, 1981.

13. Genesil System Compiler Library Volume III Random Logic Module, Silicon Compiler Corp., 1986.

14. Triebel, W.A., Integrated Digital Electronics, 2nd ed., p. 233, Prentice-Hall, 1985.

15. Conradi, J.R., and Hauenstein, B.R., VLSI Design of a Very Fast Pipelined Carry Look Ahead Adder, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1983.

16. Simchik, R.J., VLSI Design of a Sixteen Bit Pipelined Multiplier Using Three Micron NMOS Technology, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1985.

17. Taub, H., Digital Circuits and Microprocessors, p. 208, McGraw-Hill, 1982.

18. Froede, A.O. III, Silicon Compiler Design of Combinational and Pipelined Adder Integrated Circuits, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1985.

19. Waser, S., and Flynn, M.J., Introduction to Arithmetic For Digital System Designers, pp 132-138, CBS College Publishing, 1982

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Code 62 Naval Postgraduate School Monterey, California 93943-5000	1
4.	Curricular Officer, Code 32 Naval Postgraduate School Monterey, California 93943-5000	1
5.	Prof. H. H. Loomis Jr., Code 62Lm Naval Postgraduate School Monterey, California 93943-5000	1
6.	Prof. Chyan Yang, Code 62Ya Naval Postgraduate School Monterey, California 93943-5000	1
7.	LTCOL Robert H. Settle 18501 Serrano Street Villa Park, California 92667	1
8.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, D.C. 20360-0001	1
9.	Commander Space and Naval Warfare Command PD-151 Washington, D.C. 20360	1
10.	Naval Developmental Test Service Test Officer Joint Test Organization, Joint Tactical Command, Control, and Communication Agency Fort Huachuca, Arizona 85613	1
11.	Curricular Officer, Code 32 Naval Postgraduate School Monterey, California 93943-5000	1

12. Mr. Gary Harmon 1
Silicon Compiler Systems Corporation
2045 Hamilton Avenue
San Jose, California 95125

323 - 899

Thesis
S41862 Settle
c.1 Design methodology using the Genesil Silicon Compiler.

AMERICAN BINDERIES 3

thesS41862
Design methodology using the Genesil Sil
3 2768 000 84694 3
DUDLEY KNOX LIBRARY